UNIVERSIDADE DO VALE DO ITAJAÍ

UM MÓDULO DE GERÊNCIA DE REQUISITOS PARA A PLATAFORMA PLACES

Área: Engenharia de Software

Eduardo Miller

São José

2006

EDUARDO MILLER

UM MÓDULO DE GERÊNCIA DE REQUISITOS PARA A PLATAFORMA PLACES

Trabalho de Conclusão de Curso apresentado à banca examinadora do Curso de Ciência da Computação na Universidade do Vale do Itajaí - UNIVALI, São José, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Marcello Thiry

São José

2006

RESUMO

A indústria de software vem demonstrando crescente interesse no processo de gerência de requisitos, isto é, controlar o que o software se propões a fazer durante todo seu ciclo de vida. Este interesse justifica-se pelo reflexo direto deste processo em questões importantes como prazo e o custo de um projeto de software. Ao mesmo tempo em que há uma preocupação com a qualidade dos sistemas desenvolvidos, identifica-se um reflexo entre a qualidade e o gerenciamento do processo utilizado para o desenvolvimento dos sistemas e a qualidade dos sistemas desenvolvidos. Por outro lado, estudos mostram uma deficiência considerável nesta área em empresas brasileiras. Neste sentido, este trabalho apresenta uma proposta de ferramenta para apoiar o processo de gerência de requisitos, que é desenvolvida como um módulo da plataforma PLACES. Da mesma forma, a proposta desta ferramenta tem uma preocupação em estar alinhada ao modelo de melhoria de processos CMMI (*Capability Maturity Model Integration*). Para apoiar a proposta da ferramenta e verificar seu alinhamento com o modelo de melhoria, são avaliadas duas ferramentas comerciais e o próprio PLACES. Esta avaliação permitiu verificar o alto grau de maturidade das ferramentas comerciais e também serviu como referência para a identificação das melhorias que serão propostas ao PLACES. A avaliação, que foi reaplicada após a implementação das melhorias, servirá, também, para comparar e medir a evolução do PLACES. O foco deste trabalho é a melhoria do PLACES, para que ofereça recursos para apoiar o processo de gerência de requisitos.

Palavras Chaves: Gerência de Requisitos, CMMI, Ferramentas CASE

LISTA DE FIGURAS

Figura 1 - Níveis de Maturidade do CMMI. 26

Figura 2 - Perfil de Maturidade de Processos das organizações. 27

Figura 3 - Características e sub-características da norma ISO/IEC 9126. 30

Figura 4 - Etapas da Engenharia de Requisitos. 33

Figura 5 - Exemplo de caso de uso. 38

Figura 6 - Padrão do documento de Especificação dos Requisitos do Sistema. 40

Figura 7 - Detalhamento do requisito. 40

Figura 8 - Atividades da gerência de requisitos. 46

Figura 9 - Modelo do documento simplificado de solicitação de alteração de requisitos. 49

Figura 10 - Processo de controle de alterações. 51

Figura 11 - *Template* de um exemplo de Solicitação de Mudanças de Requisitos. 52

Figura 12 - Rastreabilidade do requisito. 54

Figura 13 – Exemplo de Matriz de rastreabilidade entre requisitos e casos de uso. 55

Figura 14 - Matriz de rastreabilidade com indicação de inconsistência entre os artefatos. 55

Figura 15 - Árvore de rastreabilidade ente requisitos e artefatos do sistema. 56

Figura 16 - Extrato da planilha de avaliação utilizada pelo SCAMPI. 60

Figura 17 - Regras de pontuação utilizadas pelo SCAMPI. 62

Figura 18 - Regras de pontuação utilizadas pelo SCAMPI para áreas de processo. 63

Figura 19 - Tela de associação de rastreabilidade e matriz de rastreabilidade. 66

Figura 20 - Checklist customizado na ferramenta Borland CaliberRM. 67

Figura 21 - *Template* para especificação de casos de uso no Rational RequisitePro. 68

Figura 22 - Interface da ferramenta Borland CaliberRM. 70

Figura 23 - Criação de um tipo de requisito CaliberRM. 71

Figura 24 - O Requisito criado sendo utilizado no projeto. 71

Figura 25 - Criação de usuário no CaliberRM. 72

Figura 26 – Criação de grupo de usuários no CaliberRM. 72

Figura 27 - Definição das associações de rastreabilidade do requisito no CaliberRM. 73

Figura 28 - Integração dos requisitos com tarefa do projeto, controlado no MSProject. 74

Figura 29 - Matriz de rastreabilidade do CaliberRM. 75

Figura 30 - Diagrama de rastreabilidade do CaliberRM. 75

Figura 31 - Criação de glossários no CaliberRM. 76

Figura 32 - Criação de termos no glossário no CaliberRM. 76

Figura 33 - Tela de discussões sobre o entendimento dos requisitos no CaliberRM. 77

Figura 34 - Tela de atribuição de responsabilidades dos requisitos no CaliberRM. 78

Figura 35 - Criação de *baselines* no CaliberRM. 79

Figura 36 - Template de documento do CaliberRM. 80

Figura 37 - Interface do Word com integração ao RequisitePro. 86

Figura 38 - Interface da ferramenta RequisitePro. 87

Figura 39 - Propriedades do requisito no RequisitePro. 88

Figura 40 - Propriedades do usuário no RequisitePro. 88

Figura 41 - Recursos de discussões do RequisitePro. 89

Figura 42 - *Baseline Manager* do RequisitePro. 90

Figura 43 - Definição da rastreabilidade do requisito. 91

Figura 44 - Matriz de atributos dos requisitos. 91

Figura 45 - Matriz de rastreabilidade dos requisitos. 92

Figura 46 - Árvore de rastreabilidade dos requisitos. 92

Figura 47 - Interface da plataforma PLACES. 99

Figura 48 - Tarefas do projeto no PLACES. 99

Figura 49 - Cadastramento de casos de uso no PLACES. 100

Figura 50 - Tela de projetos do PLACES. 104

Figura 51 - Listagem dos requisitos cadastrados. 106

Figura 52 - Tela de cadastro de um novo requisito. 107

Figura 53 - Detalhes de um requisito cadastrado. 108

Figura 54 - Listagem de requisitos para associação 109

Figura 55 – Área dos requisitos associados. 109

Figura 56 - Exclusão de requisito. 110

Figura 57 - Diagrama de Seqüência Cadastro de Requisitos 112

Figura 58 - Diagrama de Seqüência Associação entre Requisitos 113

Figura 59 - Lista de baselines. 113

Figura 60 - Tela de cadastro de uma nova *baseline.* 114

Figura 61 - Detalhes de uma baseline. 115

Figura 62 - Requisitos da *baseline.* 115

Figura 63 - Requisitos para inclusão na *baseline.* 116

Figura 64 - Exclusão de baselines. 117

Figura 65 - Diagrama de Seqüência Cadastro de Baselines 119

Figura 66 - Diagrama de Seqüência Inclusão de requisito em uma *baseline* 119

Figura 67 - Matriz de rastreabilidade do PLACES. 121

Figura 68 - Rastreabilidade vista na matriz de rastreabilidade. 122

Figura 69 - Diagrama de Seqüência Matriz de Rastreabilidade 123

Figura 70 - Lista de documentos anexos ao projeto. 124

Figura 71 - Adicionar um documento anexo ao projeto. 124

Figura 72 - Diagrama de Seqüência Anexar Documentos 125

Figura 73 - Divisão MVC no PLACES. 126

Figura 74 - Comparativo entre versões do PLACES. 144
Figura 75 - Tela Sobre do PLACES com módulo de Gerência de Requisitos. 145
Figura 76 - Tela de instalação do JAVA 150
Figura 77 - Tela de personalização da instalação do JAVA 150
Figura 78 - Tela de configuração das ferramentas do JAVA 150
Figura 79 - Tela de configuração do browser 151
Figura 80 - Tela de finalização da instalação 151
Figura 81 - Configuração das variáveis de Ambiente 1 152
Figura 82 - Configuração das variáveis de Ambiente 2 152
Figura 83 - Configuração das variáveis de Ambiente 3 152
Figura 84 - Configuração das variáveis de Ambiente 4 153
Figura 85 - Configuração da variável PATH 1 153
Figura 86 - Configuração da variável PATH 2 153
Figura 87 - Criar a variável JAVA_HOME 1 154
Figura 88 - Criar a variável JAVA_HOME 2 154
Figura 89 - Tela de inicio da instalação do Tomcat 154
Figura 90 - Termo de licença do Tomcat 155
Figura 91 - Configuração do Tomcat 155
Figura 92 - Configuração do local de instalação do Tomcat 1 155
Figura 93 - Tela de configuração do Tomcat 2 156
Figura 94 - Tela de configuração do Java no Tomcat 156
Figura 95 - Tela de termino da instalação do Tomcat 156
Figura 96 - Pagina principal do Tomcat 157
Figura 97 - Inicio da instalação do MySQL 157
Figura 98 - Tela de informação do MySQL 157
Figura 99 - Escolha do diretório do MySQL 158

Figura 100 - Configuração do MySQL 158
Figura 101 - Termino da instalação do MySQL 158
Figura 102 - Executar o MySQL 158
Figura 103 - Executar o MySQL 159
Figura 104 - Licença do *driver* ODBC do MySQL 159
Figura 105 - Tela de boas vindas do ODBC MySQL 159
Figura 106 - Tela de confirmação 160
Figura 107 - Tela de finalização da instalação do ODBC MySQL 160
Figura 108 - Adicionando a fonte de dados MySQL 1 160
Figura 109 - Adicionando a fonte de dados MySQL 2 161
Figura 110 - Adicionando a fonte de dados MySQL 3 161

LISTA DE TABELAS

Tabela 1 - Áreas de Processo do modelo CMMI. 25

Tabela 2 - Níveis de capacidade do CMMI. 25

Tabela 3 - Exemplo de *checklist* de verificação dos requisitos. 42

Tabela 4 - Exemplo de *checklist* de validação dos requisitos. 43

Tabela 5 - Tipos de indicadores de implementação de práticas. 59

Tabela 6 - Planilha de avaliação de ferramentas de gerência de Requisitos. 65

Tabela 7 - Colunas removidas da planilha de avaliação. 66

Tabela 8 - Componentes da ferramenta CaliberRM. 69

Tabela 9 - Planilha de avaliação da ferramenta CaliberRM. 84

Tabela 10 - Planilha de avaliação da ferramenta Rational RequisitePro. 96

Tabela 11 - Comparativo entre as ferramentas CaliberRM e RequisitePro. 98

Tabela 12 - Planilha de avaliação da ferramenta PLACES. 103

Tabela 13 - Caso de Uso Cadastro de Requisitos. 111

Tabela 14 - Caso de Uso Associação entre requisitos. 112

Tabela 15 - Caso de Uso Cadastro de Baselines. 118

Tabela 16 - Caso de Uso Inclusão de requisitos em uma *baseline.* 118

Tabela 17 - Caso de Uso Matriz de Rastreabilidade. 122

Tabela 18 - Caso de Uso da opção anexar documento. 125

Tabela 19 - Arquivo de configuração do Spring. 128

Tabela 20 - Classe ControladorAction. 128

Tabela 21 - Arquivo de configuração do hibernate. 129

Tabela 22 - Exemplo de HQL. 130

Tabela 23 - Arquivo hibernate.cfg.xml. 130

Tabela 24 - Arquivo de mapeamento objeto-relacional do *hibernate.* 132

Tabela 25 - Classe View Helper. 134

Tabela 26 - Arquivo exibeComplexidadeRequisito.tag. 135

Tabela 27 - Extrato da página addRequisito.jsp. 135

Tabela 28 - Classe ControladorRequisito.java. 136

Tabela 29 – Implementação Classe ArtefatoTO.java. 138

Tabela 30 - Planilha de reavaliação da ferramenta PLACES. 142

Tabela 31 - Comparativo entre as versões do PLACES. 144

LISTA DE ABREVIATURAS

CMM: *Capability Maturity Model*

CMMI: *Capability Maturity Model Integration*

CMMI-SW: *Capability Maturity Model Integration for Software*

CMU: *Carnegie Mellon University*

GG: *Generic Goal* – objetivo genérico

GP: *Generic Practice* – prática genérica

PA: *Process Area* – Área de processo

SEI: *Software Engineering Institute*

SG: *Specific Goal* – objetivo específico

SP: *Specific Practice* – prática específica

SCAMPI: *Standard CMMI Appraisal Method for Process Improvement*

JSP: *Java Server Pages*

MVC: *Model-View-Controller*

SUMÁRIO

LISTA DE FIGURAS **4**

LISTA DE TABELAS **9**

LISTA DE ABREVIATURAS **11**

SUMÁRIO **12**

1 INTRODUÇÃO **15**

1.1 APRESENTAÇÃO 15

1.2 OBJETIVOS 17

1.2.1 Objetivo geral *17*

1.2.2 Objetivos específicos *18*

1.3 DELIMITAÇÃO DO ESCOPO 18

1.4 JUSTIFICATIVA 19

1.5 METODOLOGIA 20

1.5.1 Aspectos Metodológicos *20*

1.6 ESTRUTURA DO TRABALHO 21

2 VISÃO GERAL DO MODELO CMMI **23**

2.1 CONCEITO GERAL 23

3 ENGENHARIA DE REQUISITOS **28**

3.1 INTRODUÇÃO 28

3.2 CONCEITO DE REQUISITO DE SOFTWARE 28

3.3 PROCESSO DA ENGENHARIA DE REQUISITOS 32

3.3.1 Definição do processo *32*

3.3.2 Elicitação de Requisitos *34*

3.3.3 Modelagem e Documentação de Requisitos *37*

3.3.4 Análise de Requisitos *41*

3.4 CONCLUSÕES DO CAPÍTULO 44

4 GERÊNCIA DE REQUISITOS **45**

4.1 INTRODUÇÃO 45
4.2 PROCESSO DE GERÊNCIA DE REQUISITOS 46
4.3 CONTROLE DE ALTERAÇÕES 48
4.4 RASTREABILIDADE 53
4.5 CONCLUSÕES DO CAPÍTULO 57

5 AVALIAÇÃO DAS FERRAMENTAS 58

5.1 INTRODUÇÃO 58
5.2 PROCESSO DE AVALIAÇÃO 58
5.3 AVALIAÇÃO DA FERRAMENTA CALIBERRM 69
5.3.1 Descrição da Ferramenta 69
5.3.2 Aplicação da avaliação 81
5.3.3 Resultados obtidos 84
5.4 AVALIAÇÃO DA FERRAMENTA REQUISITEPRO 84
5.4.1 Descrição da Ferramenta 84
5.4.2 Aplicação da avaliação 93
5.4.3 Resultados obtidos 96
5.5 AVALIAÇÃO DA FERRAMENTA PLACES 98
5.5.1 Descrição da Ferramenta 98
5.5.2 Aplicação da avaliação 101
5.5.3 Resultados obtidos 103

6 MÓDULO DE GERÊNCIA DE REQUISITOS 104

6.1 INTRODUÇÃO 104
6.2 O MÓDULO DE GERÊNCIA DE REQUISITOS 105
6.2.1 Cadastro de requisitos 106
6.2.2 Cadastro de Baselines 113
6.2.3 Matriz de Rastreabilidade 120
6.2.4 Anexar documentos 123
6.3 TECNOLOGIA UTILIZADA 126
6.3.1 Ferramentas Utilizadas 126
6.3.2 O framework SPRING 127
6.3.3 O framework Hibernate 129
6.4 PADRÕES DE PROJETO ADOTADOS 132
6.4.1 Front Controller 133
6.4.2 View Helper 133
6.4.3 Composite View 134
6.4.4 Business Delegate 135
6.4.5 Data Access Object 136
6.4.6 Business Object 137

6.4.7 Transfer Object *137*
6.4.8 Dispatcher *138*
6.5 RE-AVALIAÇÃO DO PLACES 139
6.5.1 Avaliação do Módulo de Gerência de Requisitos *139*
6.5.2 Comparação entre as versões do PLACES *142*
7 CONCLUSÃO 146
7.1 TRABALHO FUTURO 148
ANEXO 1 - INSTALAÇÃO DAS FERRAMENTAS 149
ANEXO 2 – REQUISITOS DO MÓDULO DE GERÊNCIA DE REQUISITOS 162
8 BIBLIOGRAFIA 164
9 ASSINATURAS 168

1 INTRODUÇÃO

1.1 APRESENTAÇÃO

Os principais problemas que afetam os projetos de software não são tecnológicos. A principal preocupação da indústria de software refere-se à previsibilidade de prazo, de custo e a qualidade dos produtos (HAZAN, 2004).

Para um software ter qualidade, a combinação de atributos necessária para o desempenho adequado de suas funções deve ser claramente definida (WEBER, 2001). No entanto, a produção de software, na maioria das empresas, é uma atividade artesanal; programadores trabalham basicamente com a criatividade e não costumam seguir nenhum tipo de método formal ou programa para a melhoria da qualidade do seu processo de desenvolvimento (PADUAN, 2003).

Segundo o Ministério da Ciência e Tecnologia (MCT, 2005), as empresas de desenvolvimento de software, principalmente as de pequeno e médio porte, têm problemas com o seu processo de desenvolvimento de produto o que afeta de forma negativa sua competitividade e produtividade, podendo comprometer o seu crescimento, a qualidade de seus produtos e até mesmo sua sobrevivência no mercado. Considerando que 85% do setor de informática brasileiro compõem-se de empresas de pequeno e médio porte, este problema tem chamado a atenção do MCT (MCT, 2005).

Ainda, segundo o MCT, estas empresas, em geral, têm apresentado certa deficiência organizacional e administrativa, além de uma escassez de recursos financeiros e humanos, passíveis de serem aplicados nas áreas de gestão organizacional e de processos de melhoria de software, o que implica numa dificuldade maior para a aplicação de um modelo de melhoria dos processos de software. Como conseqüência, estas empresas geralmente apresentam processos informais, não utilizando conceitos da engenharia de software, ou ainda utilizando estes conceitos de forma incorreta.

Desta forma, identifica-se um desafio que muitas organizações estão enfrentando: construir um sistema em tempo hábil para atender aos negócios com qualidade e custos adequados à sua importância e à realidade do cliente. Nos anos 90, iniciaram-se esforços orientando a solução de problemas de construção de software pela melhoria do processo de desenvolvimento, defendendo que a qualidade de um produto é reflexo da qualidade e do gerenciamento do processo utilizado no desenvolvimento deste produto (PFLEEGER, 1998).

Como resposta a esse desafio surgiram normas e padrões como as séries ISO 9000:2000 (ISO/IEC, 2000), ISO/IEC 12207 (ISO/IEC, 1995), a ISO/IEC 15504 (ISO/IEC, 2003), e o modelo CMMI (*Capability Maturity Model Integration*), desenvolvido em 2001 pelo SEI (*Software Engineering Institute*) (SEI, 2002). Apesar de ser um modelo desenvolvido inicialmente para empresas de porte maior, o modelo CMMI pode ser aplicado, com algumas adaptações, em empresas de pequeno e médio porte. Isso pode ser verificado pelo crescente esforço de organizações setoriais e acadêmicas, em promover pesquisas para a criação de abordagens para implementação do modelo nestas empresas, como, por exemplo, a pesquisa que está sendo desenvolvida pelo LQPS (Laboratório de Qualidade e Produtividade de Software), da Univali e pelo número de empresas de pequeno e médio porte avaliadas pelo SEI (SEI, 2006).

O modelo CMMI fornece uma estrutura (*framework*) para melhorias na engenharia de software, engenharia de sistemas, processo de desenvolvimento e integração de produtos e gestão de fornecedores (SEI, 2002). Neste sentido, o modelo provê um guia de referência para a melhoria dos processos de software, estabelecendo critérios mínimos necessários para planejar e implementar estes processos baseado nas metas de negócio da organização.

Segundo o MCT (2005), os maiores problemas das empresas de pequeno e médio porte referem-se à definição de custos e prazos nos projetos. Um dos principais fatores que podem gerar este problema é a deficiência no gerenciamento dos requisitos dos projetos (HAZAN, 2004). Considerando esta dificuldade este trabalho tem como foco a área de processo de "Gerência de Requisitos" do modelo CMMI. Segundo Young (2004), uma abordagem mais disciplinada para a gerência de requisitos é necessária para melhorar índices de êxito dos projetos.

O processo de Gerência de Requisitos faz parte do tema Engenharia de Requisitos. A Engenharia de Requisitos é um processo tradicional da Engenharia de Software cujo objetivo

é identificar, analisar, documentar e gerenciar os requisitos para o sistema a ser desenvolvido (PAETSCH, 2003). O modelo CMMI atende este processo por uma área de processo chamada Gerência de Requisitos, que visa estabelecer um entendimento comum entre o cliente e o fornecedor do software quanto aos requisitos, já definidos anteriormente, que serão atendidos pelo projeto, descrevendo as atividades que deverão ser executadas para obter e controlar as mudanças de requisitos e assegurar a atualização dos planos do projeto e dados relevantes (SEI, 2002).

Neste contexto, este trabalho tem como finalidade propor melhorias à plataforma PLACES, desenvolvido pelo LQPS (Laboratório de Qualidade e Produtividade de Software) da Univali, visando a implementação de um módulo para apoio ao processo de gerência de requisitos alinhado ao modelo CMMI. Para isso é feita, inicialmente, uma avaliação sobre ferramentas que poderão atender às necessidades definidas pela área de processo "Gerência de Requisitos", verificando seu alinhamento com as práticas propostas pelo CMMI. Serão avaliadas duas ferramentas comerciais e o PLACES de forma a ter uma referência da situação atual para discussão posterior. A avaliação das ferramentas comerciais também trará subsídios para a proposta de melhorias ao PLACES onde serão considerados os recursos mais importantes apresentados por estas. Após a implementação das melhorias propostas, será reaplicada à avaliação sobre a plataforma PLACES de forma a reavaliar seu alinhamento com o modelo CMMI.

Desta forma, as empresas interessadas na melhoria de processos na área de gerência de requisitos, que estão avaliando ferramentas, poderão utilizar os resultados da avaliação para melhorar seus critérios de escolha, bem como buscar, no próprio PLACES, mais uma referência de ferramenta para esta escolha.

1.2 OBJETIVOS

1.2.1 Objetivo geral

Este trabalho visa a implementação de um módulo de gerência de requisitos para a plataforma PLACES que atenda as práticas definidas pelo modelo CMMI-SW em sua versão 1.1.

1.2.2 Objetivos específicos

- Estabelecer um processo de avaliação com base no método oficial do CMMI, o SCAMPI, identificando os critérios para esta avaliação, em conformidade com a área de processo de Gerência de Requisitos do CMMI (objetivo 1).
- Aplicar a avaliação em três ferramentas de Gerência de Requisitos, sendo uma delas a plataforma PLACES, verificando seu alinhamento com o modelo CMMI (objetivo 2).
- Estabelecer um plano de melhorias à plataforma PLACES para a criação de um módulo de gerência de requisitos buscando seu alinhamento com esta área de processo do CMMI (objetivo 3).
- Desenvolver o módulo de gerência de requisitos proposto para a plataforma PLACES (objetivo 4).
- Reaplicar a avaliação na plataforma PLACES, comparando os resultados, de forma a avaliar a evolução de seu alinhamento com a área de processo de Gerência de Requisitos (objetivo 5).

1.3 DELIMITAÇÃO DO ESCOPO

Este trabalho apresenta uma visão geral do modelo CMMI, enfatizando a área de processo de Gerência de Requisitos, também buscando alguns elementos da área de processo Definição de Requisitos. Estes elementos, tais como a elicitação e a análise de requisitos, mesmo não sendo foco do trabalho, contribuirão para um melhor entendimento do processo de Gerência de Requisitos.

O foco deste trabalho é o desenvolvimento de um módulo que apóie o processo de gerência de requisitos para a plataforma PLACES. Para subsidiar este módulo, será feita uma avaliação em ferramentas de Gerência de Requisitos comerciais com a finalidade de verificar se estas possuem recursos que atendam as práticas sugeridas pelo CMMI. Com base nesta avaliação, serão propostas melhorias ao PLACES para o desenvolvimento de um módulo de gerência de requisitos que atenda as práticas do CMMI. Para apoiar esta implementação, serão explorados os principais elementos do processo de gerência de requisitos, suas etapas, técnicas utilizadas e artefatos típicos.

Da mesma forma, o plano de melhorias que será proposto à plataforma PLACES manterá o escopo nos recursos que dão suporte ao processo de gerência de requisitos sem influenciar qualquer outra funcionalidade.

1.4 JUSTIFICATIVA

Em 2001, com o surgimento da necessidade de manter o nível de competitividade, foi detectado um aumento das empresas de software que têm buscado melhorar a qualidade de seus processos (MCT, 2005). Em 2005, a mesma pesquisa, detectou que, das 15540 empresas e prestadores de serviço no setor de software, 3404 situam-se na região sul e, uma parcela considerável, aproximadamente 50%, ainda não possuía qualquer programa de qualidade visando melhorias no processo de desenvolvimento de software (MCT, 2005).

Segundo o CHAOS Report – 2003, do Standish Group, a indústria continua lidando com projetos mal sucedidos, conforme as estatísticas: 34% dos projetos são bem sucedidos; 15% dos projetos foram cancelados; 43% é o erro médio com relação ao orçamento do projeto daqueles que foram completados; 52% das características (requisitos não funcionais) e funcionalidades são entregues no produto. Os dados desta pesquisa mostram que a maior parte dos problemas encontrados está relacionada aos requisitos, tanto no processo de definição quanto no processo de gerenciamento dos mesmos (STANDISH GROUP *apud* HAZAN, 2004).

Como o custo para corrigir os problemas encontrados após a fase de testes é, até 20 vezes maior que o custo de correção para os problemas encontrados na fase de definição dos requisitos, o gerenciamento de requisitos tem sido uma das maiores preocupações da indústria de software (SAYÃO, 2004). Os dois principais fatores que contribuem para falhas de projetos de desenvolvimento de software em atender seus orçamentos e cronograma são problemas de requisitos: especificação de requisitos inadequada e mudanças em requisitos (HAZAN, 2004). Assim, as organizações que aplicarem uma gerência de requisitos durante o ciclo de vida do projeto, podem aumentar ao máximo os benefícios provenientes desta melhoria de processo pelo controle de mudanças de requisitos.

Neste sentido, a utilização da avaliação que é executada neste trabalho auxiliará as empresas a escolher a ferramenta que melhor atenda às necessidades da área de gerência de requisitos, de forma a controlar melhor a relação crítica entre requisitos, custos e prazos. Da mesma forma,

as melhorias que serão desenvolvidas na plataforma PLACES oferecerá mais recurso gratuito e *open source* às empresas que tiverem interesse em melhorar sua área de gerência de requisitos.

1.5 METODOLOGIA

1.5.1 Aspectos Metodológicos

Segundo Silva (2001), existem duas formas clássicas de classificação de pesquisas quanto à sua natureza, a pesquisa básica, que objetiva criar novos conhecimentos para a ciência e a pesquisa aplicada que objetiva criar conhecimentos para a aplicação prática e dirigidos a solucionar problemas específicos geralmente de interesse local. De acordo com essa classificação, este trabalho é de natureza aplicada, uma vez que será adaptado um método para avaliação de ferramentas utilizando uma base de conhecimento existente, o modelo CMMI.

Já do ponto de vista da forma de abordagem, este estudo é classificado como qualitativo uma vez que não será necessária a utilização de métodos estatísticos. Este trabalho será descritivo e seus dados serão analisados indutivamente com base nos critérios de avaliação que serão discutidos a partir dos objetivos específicos da área de processo de gerência de requisitos definidos pelo modelo CMMI. Para Andrade (1999), o método indutivo privilegia a observação como processo para chegar-se ao conhecimento e consiste em enumerar os enunciados sobre o fenômeno que se pretende pesquisar e, através da observação procura-se encontrar algo que está sempre presente na ocorrência do fenômeno.

Do ponto de vista de seus objetivos, será adotada uma pesquisa exploratória. Ela visa proporcionar maiores informações sobre um determinado assunto para torná-lo explícito ou para permitir a construção de hipóteses (RICHARDSON *et al*, 1999, p.45). A pesquisa exploratória compreende levantamento bibliográfico e análise de exemplos que estimulem a compreensão. Ela assume, em geral, as formas de pesquisas bibliográficas e estudos de caso (ANDRADE, 1999, p.106).

Para a execução do trabalho, serão adotados os procedimentos técnicos de pesquisa bibliográfica e pesquisa-ação. A pesquisa bibliográfica procura explicar um problema a partir de referências teóricas publicadas em documentos (CERVO; BERVIZ, 1996, p.48), constituído principalmente de livros, artigos de periódicos e atualmente com material

disponibilizado na Internet. Neste sentido, este trabalho apresentará um estudo aprofundado sobre o processo de gerência de requisitos. Serão estudadas as atividades relevantes deste processo, bem como técnicas utilizadas que poderiam ser utilizadas no contexto do trabalho.

A pesquisa-ação é caracterizada quando o trabalho é concebido e realizado em estreita associação com uma reação ou com a resolução de um problema coletivo. Os pesquisadores e participantes representativos da situação ou do problema estão envolvidos de modo cooperativo ou participativo (GIL, 1999, p.46). É dentro deste contexto que este trabalho proporá um plano de melhorias à plataforma PLACES, alinhando-a ao modelo CMMI, com base nos resultados obtidos a partir da avaliação executada. Após a implementação do plano de melhorias, será reaplicada a avaliação e os resultados serão discutidos, verificando se as melhorias implementadas tornaram o PLACES alinhado ao CMMI.

1.6 ESTRUTURA DO TRABALHO

O trabalho está organizado em 5 capítulos correlacionados, iniciando com esta introdução sobre o tema proposto onde foi apresentado o tema a ser abordado neste trabalho por meio de sua contextualização. Da mesma forma, foram estabelecidos os resultados esperados por meio da definição de seus objetivos e apresentadas as limitações do trabalho permitindo uma visão clara do escopo proposto.

O capitulo dois apresenta uma visão sobre a Engenharia de Requisitos, suas técnicas e abordagens. Esta conceituação tem como objetivo estabelecer uma referência sobre o processo de definição de requisitos, incluindo as atividades de elicitação, modelagem e análise. Estes conceitos serão utilizados para subsidiar o entendimento das práticas do CMMI para a aplicação da avaliação das ferramentas.

O capitulo três aborda o processo de gerência de requisitos, apresentando suas atividades de controle de alterações e manutenção da rastreabilidade entre os requisitos e os demais artefatos do projeto. Em função do escopo deste trabalho, optou-se por criar um capítulo para tratar exclusivamente este tema, enfatizando suas atividades com um maior detalhamento. Também será abordado, neste capítulo, o modelo CMMI e sua área de processo de Gerência de Requisitos, mostrando seus objetivos relacionando-os aos processos abordados.

No capítulo quatro é apresentada a avaliação das ferramentas de gerência de requisitos tendo como referência o método de avaliação oficial do CMMI, o SCAMPI. Serão avaliadas duas

ferramentas comerciais, o Borland CaliberRM e o Rational RequisitePro e a plataforma PLACES. O resultado das avaliações será discutido e a avaliação das ferramentas comerciais servirá como fonte de referência para as melhorias que serão implantadas no módulo que será desenvolvido no PLACES.

O capítulo cinco aborda a atividade que é o objetivo principal deste trabalho, o desenvolvimento do Módulo de Gerência de Requisitos para a plataforma PLACES. Neste mesmo capítulo é apresentado o Módulo desenvolvido junto com sua documentação e a segunda avaliação do PLACES, executada após a implementação do novo módulo. Os resultados desta segunda avaliação são mostrados e discutidos de forma a avaliar a evolução do alinhamento do PLACES com o modelo CMMI.

2 VISÃO GERAL DO MODELO CMMI

2.1 CONCEITO GERAL

O CMMI é um *framework* que acomoda as disciplinas de engenharia de sistemas, engenharia de software, desenvolvimento de integração de processos e produtos e subcontratação (fornecedor/aquisição). Sua proposta é prover um guia para a melhoria de processos da organização e a capacidade de gerenciar o desenvolvimento, aquisição de produtos e serviços (SEI, 2002).

Inicialmente, para facilitar a utilização de acordo com as necessidades, foram desenvolvidos pela Universidade Carnegie Mellon quatro documentos do CMMI, disponíveis no site do SEI (SEI, 2002):

- CMMI-SW (*CMMI for Software Engineering*): para a utilização individual da disciplina de Engenharia de Software;
- CMMI-SE (*CMMI for Systems Engineering*): para a utilização individual da disciplina de Engenharia de Sistemas;
- CMMI-SE/SW *(CMMI for Systems Engineering and Software Engineering*): para a utilização conjunta das disciplinas de Engenharia de Software e Engenharia de Sistemas;
- CMMI-SE/SW/IPPD (*CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development and Supplier Sourcing*): para a utilização conjunta das disciplinas de Engenharia de Software, Engenharia de Sistemas e Desenvolvimento Integrado de Produtos e Processos.

O *framework* CMMI é a estrutura básica que organiza os componentes, incluindo elementos comuns dos modelos CMMI, como também, métodos de avaliação (incluindo artefatos associados), material de treinamento e regras e métodos para a geração de modelos. O modelo possui diversos componentes. Os principais componentes que fazem parte do modelo são:

- Áreas de processo (PA – *Process Area*): são conjuntos de práticas relacionadas em uma área que, quando executadas coletivamente, satisfazem um conjunto de objetivos considerados importantes para a promoção de melhorias naquela área.
- Objetivos específicos (SG – *Specific Goal*): são objetivos que se aplicam a cada área de processo e tratam de características únicas que descrevem o que deve ser realizado para satisfazer a área de processo. Os objetivos específicos são componentes exigidos do modelo e são utilizados nas avaliações para auxiliar a determinar se a área de processo está sendo satisfeita.
- Práticas específicas (SP – *Specific Practice*): são atividades que são consideradas importantes para atender a um objetivo específico. As práticas específicas descrevem as atividades que se espera que a organização execute para atender os objetivos específicos de uma área de processo. As práticas específicas são componentes esperados do modelo e não são exigidos numa avaliação.
- Objetivos genéricos (GG – *Generic Goal*): são considerados genéricos por serem objetivos associados a todas as áreas de processo. Quando um objetivo genérico é atingido em uma área de processo, isto significa um controle melhorado no planejamento e execução dos processos associados àquela área de processo.
- Práticas genéricas (GP – *Generic Practice*) – são atividades que fornecem institucionalização para garantir que os processos associados à área de processo serão efetivos, repetíveis e duráveis.

Na tabela 3 são listadas as áreas de processo juntamente com seus objetivos específicos.

Área de Processo	Objetivo específico
Gerência de Requisitos	Administrar os requisitos dos produtos e seus componentes, identificando as inconsistências entre esses requisitos e os planos dos projetos dos produtos.
Planejamento de Projeto	Definir e manter o planejamento que define as atividades de projeto.
Acompanhamento e Controle de Projeto	Fornecer uma análise do andamento do projeto de modo que ações corretivas apropriadas possam ser tomadas quando o desempenho do projeto desvia-se significativamente do previsto.
Gerência de Acordos com Fornecedores	Administrar a aquisição de produtos de maneira a estabelecer um acordo formal de fornecimento com seus respectivos fornecedores.
Garantia de Qualidade de Processo e Produto	Fornecer pessoal e gerência com ação objetiva na verificação da qualidade dos processos e produtos associados.
Gerência de Configuração	Estabelecer e manter a integridade de produtos usando uma identificação, controle, verificação, e auditorias de configuração dos mesmos.
Medição e Análise	Desenvolver a capacidade de medição usada para apoiar a necessidade de informação da gerência da organização.
Foco no Processo da Organização	Planejar e implementar melhorias nos processos da organização com base em num entendimento completo dos pontos fortes e fracos destes.
Definição do Processo da Organização	Estabelecer e manter o conjunto de processos organizacionais que são considerados como ativos.

Treinamento Organizacional	Desenvolver as habilidades e conhecimento das pessoas para que possam então executar seus papéis de maneira efetiva e eficiente.
Gerência Integrada de Projeto	Estabelecer e gerenciar o projeto e o envolvimento do pessoal relevante a este, conforme a integração prevista no planejamento dos processos definidos para este projeto.
Gerência de Risco	Identificar problemas potenciais antes que eles ocorram, de modo que as atividades de gerência de riscos possam ser planejadas e executadas durante o ciclo de vida projeto, visando, assim, minimizar impactos ao planejamento.
Desenvolvimento de Requisitos	Analisar os clientes, os produtos e seus componentes, definindo, assim seus requisitos.
Solução Técnica	Projetar, desenvolver, e implementar as soluções para os requisitos identificados. As soluções, projetos, e implementações incluem produtos, seus componentes, processos de desenvolvimento.
Integração de Produto	Montar o produto a partir de seus componentes assegurando que este, da maneira que foi integrada, funcione conforme foi previsto. Entregar o produto.
Verificação	Assegurar que os produtos atendam aos requisitos que foram especificados.
Validação	Demonstrar o produto, e seus componentes, cumpra sua finalidade quando colocado no ambiente previsto.
Análise de Decisão e Resolução	Analisar as possíveis decisões através de um processo formal de avaliação, avaliando as alternativas identificadas em relação a critérios que devem ser previamente estabelecidos.
Desempenho do Processo Organizacional	Estabelecer e manter um entendimento quantitativo do desempenho dos processos da organização em favor da qualidade e dos objetivos de melhoria destes processos. Fornecer os dados de desempenho dos processos, pontos de partidas, e modelos para administrar quantitativamente os projetos da organização.
Gerência Quantitativa de Projeto	Administrar quantitativamente os processos de um projeto com a finalidade de alcançar a qualidade e a melhoria prevista para estes.
Análise Causal e Resolução	Identificar causas de defeitos e outros problemas e tomar ações para prevenir sua ocorrência no futuro.
Inovação e Melhoria Organizacional	Selecionar e aplicar melhorias, de maneira seqüencial e inovadora, de maneira que possam melhorar os processos e as tecnologias utilizadas.

Tabela 1 - Áreas de Processo do modelo CMMI.
Fonte: SEI, 2001

O CMMI é um modelo flexível o suficiente para suportar duas representações (em estágios e contínua), uma voltada para a avaliação e definição do nível de capacidade dos processos e a outra voltada à definição do nível de maturidade da organização. É importante ressaltar que todas as áreas de processo do modelo são comuns para as duas representações. O modelo possui seis níveis de capacidade, que são aplicados à representação contínua, e cinco níveis de maturidade, que são aplicados à representação em estágios.

Desta forma, os níveis de capacidade aplicam-se a cada área de processo da organização, definindo o nível de melhoria para cada uma delas. São previstos seis níveis de capacidade, identificados de zero a cinco (SEI, 2002). Na tabela 4 estão listados os seis níveis de capacidade do CMMI.

Níveis de Capacidade
0 – Incompleto
1 – Executado
2 – Gerenciado
3 – Definido
4 – Quantitativamente Gerenciado
5 – Otimizado

Tabela 2 - Níveis de capacidade do CMMI.
Fonte: SEI, 2002

Já os níveis de maturidade, que pertencem à representação em estágios, têm a finalidade de avaliar a maturidade da organização, de forma que são aplicados a grupos de áreas de

processo. O modelo possui cinco níveis de maturidade, numerados de um a cinco (SEI, 2002). Quando a organização alcança os objetivos em todas as áreas de processo definidas para um nível de maturidade, ela alcança este nível de maturidade. Na figura 7, se pode visualizar os níveis de maturidade juntamente com as áreas de processo (PAs) que fazem parte de cada um.

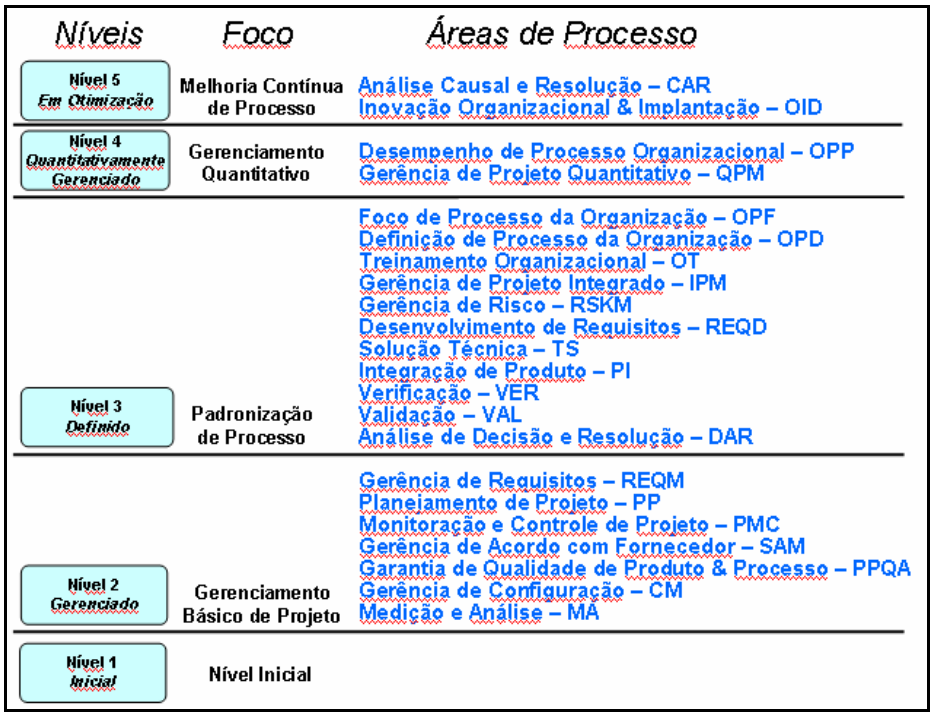

Níveis
Foco
Áreas de Processo
Nível 5
Em Otimização
Melhoria Contínua de Processo
Análise Causal e Resolução – CAR
Inovação Organizacional & Implantação – OID
Nível 4
Quantitativamente Gerenciado
Gerenciamento Quantitativo
Desempenho de Processo Organizacional – OPP
Gerência de Projeto Quantitativo – QPM
Nível 3
Definido
Padronização de Processo
Foco de Processo da Organização – OPF
Definição de Processo da Organização – OPD
Treinamento Organizacional – OT
Gerência de Projeto Integrado – IPM
Gerência de Risco – RSKM
Desenvolvimento de Requisitos – REQD
Solução Técnica – TS
Integração de Produto – PI
Verificação – VER
Validação – VAL
Análise de Decisão e Resolução – DAR
Nível 2
Gerenciado
Gerenciamento Básico de Projeto
Gerência de Requisitos – REQM
Planejamento de Projeto – PP
Monitoração e Controle de Projeto – PMC
Gerência de Acordo com Fornecedor – SAM
Garantia de Qualidade de Produto & Processo – PPQA
Gerência de Configuração – CM
Medição e Análise – MA
Nível 1
Inicial
Nível Inicial

Figura 1 - Níveis de Maturidade do CMMI.
Fonte: SEI. 2002

Desta forma, é importante ressaltar que os níveis de maturidade são alcançados pelo atendimento dos objetivos (do nível) para todas as PAs definidas para o nível. Assim, para uma organização alcançar o nível 2 de maturidade, será necessário alcançar todos os objetivos (do nível dois) para as áreas de processo Gerência de Requisitos, Planejamento de Projeto, Monitoração e Controle de Projeto, Gerência de Acordo com Fornecedores, Garantia da Qualidade de Produto e Processo, Gerência de Configuração e Medição e Análise. Interpreta-se, desta forma, que a organização atingiu o Nível de Maturidade Gerenciado.

Caso a organização não queira aplicar o modelo usando a representação por estágios, e sim melhorar apenas alguns processos que considere mais relevante, ela pode optar pela representação continua, como visto anteriormente. Desta forma, a empresa escolhe as áreas de processo de seu interesse para aplicar as melhorias e, neste ponto de vista, aplica-se, a cada área de processo um nível de capacidade, indicando o nível de melhoria da área de processo. Segundo o SEI (2005), 90,2% das organizações que foram verificadas, optou por aplicar o modelo CMMI em sua representação por estágios, como mostra a figura 8.

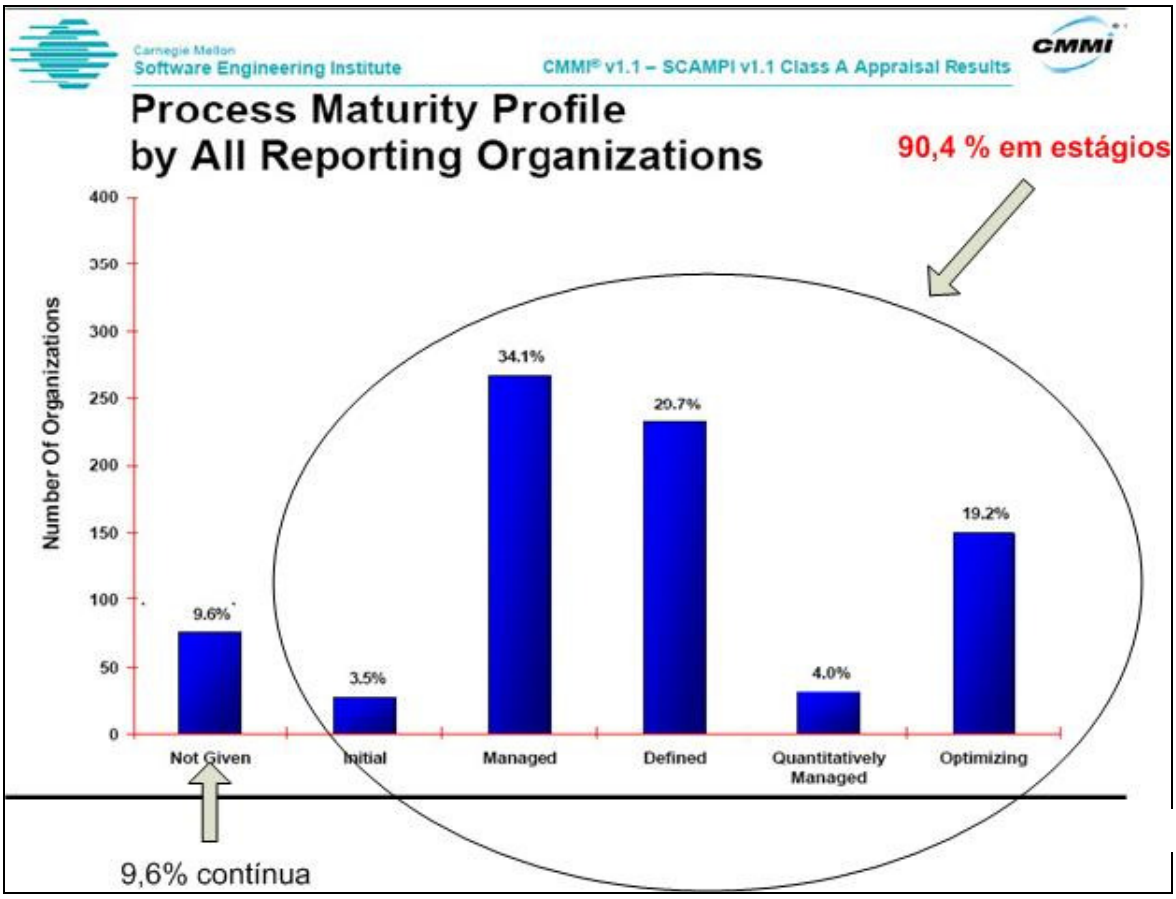


Figura 2 - Perfil de Maturidade de Processos das organizações.
Fonte: SEI, 2005

Em função da escolha detectada pela pesquisa do SEI, será adotada, neste trabalho, a visão por estágios como referência para efetuar a avaliação das ferramentas e buscar o alinhamento com o CMMI. A próxima etapa deste estudo é a apresentação das atividades do processo de gerência de requisitos. As sessões a seguir abordarão estas atividades fazendo referência às praticas que o modelo CMMI exige. Desta forma estaremos criando uma base de conhecimento para a aplicação da abordagem, foco deste trabalho.

3 ENGENHARIA DE REQUISITOS

3.1 INTRODUÇÃO

Este capítulo aborda a Engenharia de Requisitos, suas fases e artefatos típicos com o objetivo de mostrar como é feito o levantamento dos requisitos e sua documentação. Este conhecimento serve de base para o atendimento da prática de avaliação dos requisitos tratada no capítulo três, que trata da área de processo de Gerência de Requisitos do CMMI. É importante ressaltar que, em função do foco deste trabalho, o processo de gerenciamento dos requisitos será tratado em um capítulo próprio.

A Engenharia de Requisitos é um amplo campo inserido no contexto da Engenharia de Software, compreendendo um processo que envolve todas as atividades necessárias para criar e manter a documentação de requisitos do sistema. É a área de estudo que procura sistematizar o processo de definição de requisitos. Sua função principal é aperfeiçoar os processos do ciclo de vida dos requisitos e aborda um ponto fundamental do desenvolvimento de software: a definição do que produzir. A Engenharia de Requisitos tem sido identificada como uma fase crucial por tratar de conhecimentos não apenas técnicos, mas também gerenciais, organizacionais, econômicos e sociais, e estar intimamente associada à qualidade do software (CASTRO, 1995).

O produto da Engenharia de Requisitos são documentos, chamados de requisitos, que definem as premissas para o desenvolvimento do produto. Assim, a Engenharia de Requisitos constitui-se de uma seqüência de descobertas e refinamentos essencial para o projeto, ma vez que organiza uma base para a construção do software.

3.2 CONCEITO DE REQUISITO DE SOFTWARE

Antes de abordar o Processo de Engenharia de Requisitos, é importante conceituar requisito, termo freqüentemente citado, debatido e utilizado na redação de contratos, sem que as partes possuam uma compreensão única de seu significado.

Existem várias definições para requisitos, entre as quais foram escolhidas algumas para citar:

"Necessidades básicas do cliente, geralmente explicitadas como condição de negócio no contrato com o fornecedor. São características, tais como funcionalidades, especificações técnicas, prazo de entrega, garantia, que o cliente "requer" do produto."(MCT, 2005)

"Uma condição ou capacidade necessitada por um usuário, para resolver um problema ou alcançar um objetivo." (IEEE, 1983)

"Necessidades ou expectativas que são expressas, geralmente, de forma implícita ou obrigatória." (ISO/IEC 9000, 2000)

"*Os requisitos de um sistema de computação constituem uma especificação das características e propriedades do sistema ou uma descrição do que o sistema deve fazer, de como ele deve se comportar, bem como de suas restrições de operação*." (LEFFINGWELL, 2003)

É importante ressaltar que os requisitos devem dizem "o que o sistema deve fazer" e "o que o sistema não deve fazer", sem dizer "o como fazer".

Os requisitos de um sistema podem ser organizados em diferentes níveis de abstração: requisitos de negócio, requisitos de usuário e requisitos funcionais (KOTONYA E SOMMERVILLE, 1998).

Os requisitos de negócio correspondem aos objetivos de negócio que devem ser satisfeitos pelo sistema. Normalmente são descritos em um documento denominado visão ou escopo do sistema. Requisitos de usuário descrevem as atividades que os usuários deverão se capazes de executar com a utilização do sistema.

Os requisitos funcionais definem as funcionalidades que o sistema deve possuir para que os usuários possam executar suas atividades, de modo a atingir os objetivos do negócio (YOUNG, 2004). Ainda, segundo Hazan (2004), "Os requisitos funcionais são a descrição das diversas operações que clientes e usuários querem, ou precisam, que sejam realizadas pelo sistema". É importante ressaltar que não consideração desses fatores na análise de requisitos constitui uma das principais razões de uma eventual insatisfação do usuário com relação a um produto (ROCHA, 2001).

Além destes, uma especificação de requisitos deve conter os requisitos não-funcionais. Segundo Sommerville (2001), os requisitos não-funcionais são restrições que o software deve

atender ou atributos de qualidade específicas que um software ou de um processo de desenvolvimento de software deve ter. Em função desta característica, é importante que requisitos deste tipo sejam considerados na fase inicial do processo de desenvolvimento. Os requisitos não-funcionais referem-se também a questões como desempenho, segurança, rentabilidade, usabilidade e confiabilidade, entre outros, quando aplicáveis ao software.

Como referência para a identificação de requisitos não-funcionais indica-se a norma ISO/IEC 9126 (ISO/IEC *apud* REINEHR, 2006), que define as características e subcaracterísticas para qualidade de um produto de software. Verificando a necessidade do produto com base nas características citadas nesta norma, pode-se definir com maior facilidade os requisitos não-funcionais. As características e suas subcaracterísticas indicadas pela norma podem ser vistas na figura 1.

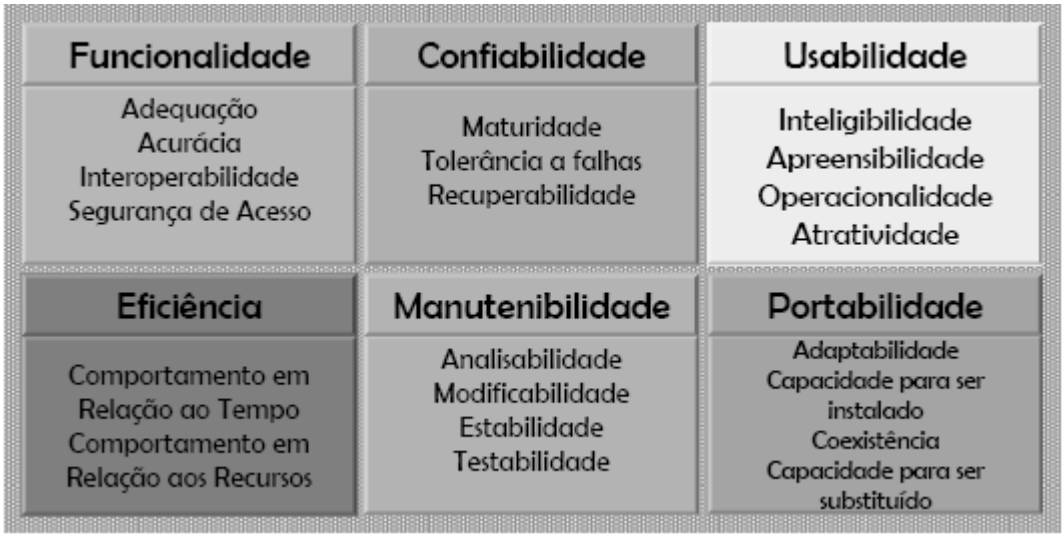


Figura 3 - Características e sub-características da norma ISO/IEC 9126.
Fonte: Reinehr *apud* ISO/IEC, 2006

Definições das características e subcaracterísticas da norma ISO/IEC 9126:

- Funcionalidade: a funcionalidade define o conjunto de atributos que evidenciam a existência de um conjunto de funções e suas propriedades específicas.
    - Adequação: o sistema tem esta subcaracterística se possui funções apropriadas para as tarefas e se há a presença das funcionalidades esperadas;
    - Acurácia: o sistema fornece resultados corretos, esperados e precisos;
    - Interoperabilidade: o sistema pode interagir com outros sistemas;
    - Segurança de acesso: há proteção de dados e controle de acesso.

- Confiabilidade: indica um conjunto de que evidenciam a capacidade do software de manter seu nível de desempenho esperado sob condições estabelecidas por um período estabelecido.
    - Maturidade: evitar falhas decorrentes de defeitos de software;
    - Tolerância à falhas: manter um nível de desempenho em caso de defeitos;
    - Recuperabilidade: capacidade de restabelecer o nível de desempenho e recuperar dados afetados por alguma falha.
- Usabilidade: indica um conjunto de atributos que evidenciam o esforço necessário para se poder usar o software, bem como o julgamento individual deste uso por um conjunto de usuários.
    - Inteligibilidade: facilita o entendimento sobre o uso do produto;
    - Apreensibilidade: fácil de aprender a usar o produto;
    - Operacionabilidade: fácil de operar e controlar o produto;
    - Atratividade: o produto é atrativo ao usuário.
- Eficiência: Conjunto de atributos que evidenciam o relacionamento entre o nível de desempenho do software e a quantidade de recursos utilizados, sob condições estabelecidas.
    - Comportamento em relação ao tempo: fornece as respostas em tempo apropriado (desempenho);
    - Comportamento em ralação aos recursos: uso racional de recursos (economia);
- Manutenibilidade: Conjunto de atributos que evidenciam o esforço necessário para fazer modificações especificadas no software.
    - Analisabilidade: medida do esforço necessário para localizar um problema ou o que precisa ser modificado para corrigi-lo;
    - Modificabilidade: habilita uma modificação específica;
    - Estabilidade: modificações não afetam o comportamento do produto;
    - Testabilidade: modificações podem ser validadas.

- Portabilidade: Conjunto de atributos que evidenciam a facilidade de um software de ser transferido de um ambiente para outro.
    - Adaptabilidade: adaptação para outros ambientes;
    - Capacidade para ser instalado: instalação em ambientes determinados;
    - Capacidade para substituir: usado no lugar de outros sistemas (por exemplo, novas versões);
    - Capacidade de coexistir: conviver com outros sistemas, compartilhando recursos.

Algumas características, como, por exemplo, a usabilidade, devem ser verificadas junto aos usuários do sistema. Este *feed-back* ajuda a verificar e validar aspectos não claramente definidos como a facilidade de operar, a facilidade de aprender ou a atratividade do sistema.

É importante ressaltar que o produto de software, conforme sua finalidade e necessidade, deve ser analisado a fim de definir que características deve ou não possuir. A verificação destas características serve como base para a identificação de requisitos, principalmente os não-funcionais.

Além da classificação em funcionais e não-funcionais, os requisitos podem ser classificados em requisitos de produto, de projeto e de processo. Os requisitos de produto são aqueles que devem, deveriam ou podem ser atendidos pelo produto entregue. Os requisitos de projeto estipulam recursos que precisam estar disponíveis e como diferentes aspectos do projeto devem ser tratados. Os requisitos de processo indicam padrões, procedimentos, métodos, linguagens, processos de engenharia e de gerenciamento que devem ser seguidos. Os requisitos de projeto e de processo são também requisitos não-funcionais e os requisitos de produto podem ser não-funcionais ou funcionais.

Antes de iniciar uma especificação de requisitos, é importante que a organização tenha definido um processo para a execução desta atividade.

3.3 PROCESSO DA ENGENHARIA DE REQUISITOS

3.3.1 Definição do processo

Para elaborar e manter uma especificação de requisitos é necessário que os desenvolvedores executem um conjunto estruturado de atividades destinadas a elicitar, documentar, analisar e

gerenciar os requisitos. Este conjunto de atividades, iterativas, recebe o nome de Processo de Engenharia de Requisitos.

Quando a organização não dispõe deste processo formalmente definido e amplamente divulgado, os desenvolvedores elaboram as especificações dos requisitos de forma empírica, executando atividades não padronizadas e definidas individualmente. Se isto ocorre, a qualidade da especificação dependerá exclusivamente da experiência e formação das pessoas, havendo assim uma elevada probabilidade de ocorrerem conflitos e re-trabalho (WIEGERS, 1999).

As atividades que compõem o Processo de Engenharia de Requisitos é um conjunto de boas práticas que apóiam sua execução Segundo Cláudia Hazan (2004) são a elicitação, a modelagem e documentação, a análise e o gerenciamento dos requisitos.

Na elicitação, deve-se entender o problema, por meio de entrevistas, questionários e análise de documentos. Em seguida faz-se a modelagem que é a representação gráfica do entendimento do problema, utilizando-se de alguma ferramenta de modelagem tal como diagramas de fluxo de dados, diagramas de uso de caso, ou outra ferramenta. Após esta etapa executa-se a análise dos requisitos obtidos pela a verificação e validação dos modelos obtidos os as informações coletadas na fase de elicitação.

Em qualquer etapa, uma vez encontrada alguma inconsistência, deve-se rever o requisito encontrado refazendo, se necessário, a modelagem ou até a elicitação do mesmo, como é ilustrado na figura 2 (HAZAN, 2004).

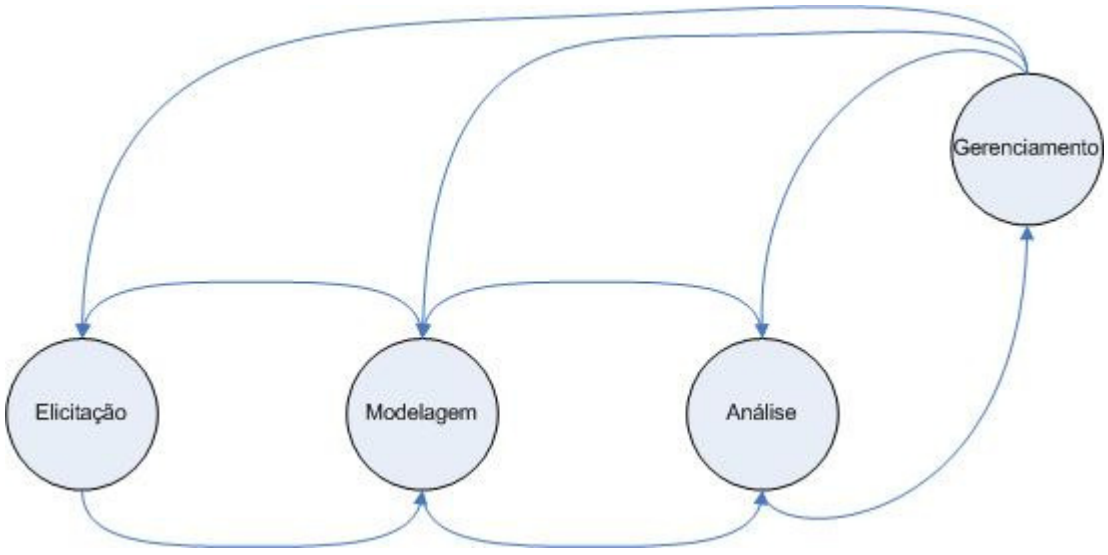


Figura 4 - Etapas da Engenharia de Requisitos.
Fonte: HAZAN, 2004

A seguir são vistas cada uma das etapas do processo, descrevendo cada atividade e citando algumas técnicas utilizadas.

3.3.2 Elicitação de Requisitos

3.3.2.1 Conceito

A elicitação de requisitos é a atividade de descobrimento dos requisitos de um sistema; a descrição de um produto de software específico. Desenvolvedores, engenheiros, colaboradores e usuários trabalham juntos para descobrir o problema a ser resolvido, os serviços do sistema, o desempenho necessário, as restrições de hardware, entre outras informações. Nesta fase ocorre o levantamento ou extração dos requisitos (KOTONYA E SOMMERVILLE, 1998). Desta forma, este processo torna-se responsável por descobrir as informações, compreender os fatos descobertos e adquirir conhecimento pela extração de informações sobre a funcionalidade requisitada e outras propriedades do sistema.

O termo elicitação de requisitos sugere que o processo seja uma simples transferência de conhecimento onde os engenheiros de requisitos fazem o levantamento e a documentação do conhecimento do cliente. Na realidade este processo é bem mais complexo, pois raramente se tem uma visão clara dos requisitos; diferentes pessoas têm requisitos conflitantes; usuários têm entendimento incompleto de suas necessidades; usuários conhecem pouco sobre a capacidade e limitações de computadores; analistas têm pouco conhecimento sobre o domínio do problema; ocorre a omissão de informações “óbvias”; requisitos freqüentemente vagos e não-testáveis; entre outras. Outro problema é a volatilidade dos requisitos, que evoluem com o tempo devido a mudanças no negócio, mudanças tecnológicas ou, até mesmo, para uma maior clareza pelo usuário do sistema. A saída do processo de elicitação de requisitos é um conjunto de esboços de documentos que representam os requisitos do sistema (SOMMERVILLE, 2001).

O produto final da etapa de elicitação pode ser um documento escrito, um modelo gráfico, uma coleção de cenários de uso, um protótipo, ou qualquer combinação dos itens citados. Portanto, ao final desta etapa é gerado um documento chamado Requisitos do Cliente, que pode ter caráter individual ou fazer parte do documento mais abrangente chamado de Especificação dos Requisitos do Sistema. Este documento pode ser escrito contendo

linguagem natural combinada a modelos gráficos. O documento de requisitos de software tem importância fundamental, pois constitui a base para as estimativas de tamanho, esforço, cronograma, e custo (HAZAN, 2004).

3.3.2.2 Técnicas

A elicitação de requisitos é a primeira atividade no Processo de Engenharia de Requisitos, onde se busca entender o domínio da aplicação, o problema, o negócio e as necessidades e restrições das pessoas envolvidas no sistema. Esta atividade passa por duas etapas: a identificação das fontes de dados e a coleta dos dados (HAZAN, 2004).

As fontes de dados podem ser: documentos, livros, sistemas legados e os atores do universo de informações pertinentes ao sistema (clientes, usuários e desenvolvedores). No momento de identificação das fontes de informação, algumas perguntas devem ser respondidas: Quem é o cliente do sistema? Quem é o dono do sistema? Quais são os documentos, referenciados pelos atores pertinentes à aplicação? Existe a possibilidade de reutilizar algum artefato de software? Existe alguma solução, ou pacote, disponível no mercado?

Após definidas as fontes de informações, passa-se a etapa de coleta dos dados. Esta coleta é feita utilizando-se diversas técnicas tais como: leitura de documentos, entrevistas, questionários, reuniões e observação. Em todas as técnicas, faz-se necessária a participação ativa dos atores do sistema (SAYAO, 2004).

A seguir é descrito as vantagens e os cuidados que se deve ter ao aplicar cada técnica citada, conforme proposto por Julio Cesar Leite (2004):

- Leitura de documentos: a leitura de documentos traz um grande volume de informações além de prover uma facilidade ao acesso às informações, porém deve-se tomar cuidado em relação à dispersão destas informações como também ao tempo necessário a esta leitura.
- Entrevistas: esta técnica traz algumas vantagens tais como o contato direto com os atores do sistema e uma possibilidade de validação imediata da informação coletada. Por outro lado, deve-se atentar para que as informações coletadas não se baseiem somente no conhecimento tácido dos atores entrevistados, o que pode dificultar a coleta de informações. O conhecimento tácito é aquele que é trivial para o ator, mas não é comum ao entrevistador e, por este motivo, alguma informação pode não ser lembrada se não for questionada pelo entrevistador.

- Reuniões: as reuniões oportunam múltiplas opiniões e a criação coletiva de requisitos. Porém, avaliar a questão dos custos que as reuniões podem trazer e como elas são direcionadas uma vez que podem gerar uma grande dispersão em torno do assunto discutido.
- Questionários: A utilização de questionários tem como vantagens a padronização das perguntas e um posterior estudo estatístico em cima das informações coletadas. Por outro lado tem-se pouca iteração com os questionados e uma possível limitação das respostas. Observar que para a elaboração de um questionário, deve-se possuir um conhecimento mínimo para possibilitar o correto direcionamento das questões.
- Prototipagem: consiste na produção de uma versão inicial do sistema, que possa estar disponível aos usuários na fase inicial do processo de desenvolvimento (KOTONYA E SOMMERVILLE, 1998). Os protótipos também podem ser produzidos como uma seqüência de telas em papel, descrevendo a funcionalidade de cada elemento com o qual o usuário poderá interagir.
- JAD (*Joint Application Design*): JAD é uma técnica de reunião que tem por objetivo acelerar e consolidar o desenvolvimento de aplicações (LOCH, 2003). Sua aplicação permite a criação de sistemas mais eficazes em menor tempo, tendo como um grande benefício à aderência que pode ter a vários métodos de desenvolvimento de software atualmente empregados (LOCH, 2003). O JAD como uma técnica orientada ao trabalho em grupo, que oferece um ambiente para clientes e engenheiros de requisitos trabalharem como um time, compartilhando informações e idéias sobre um determinado assunto. Faz uso de sessões de *brainstorming*[1] mediada por um facilitador, que assumirá o papel de conduzir as discussões e mediar conflitos durante a sessão. À medida que as discussões progridem, os requisitos que vão sendo levantados deverão ser transcritos pelo facilitador ou um auxiliar (WOOD, 2005).

[1] Técnica que envolve a geração e redução de idéias. Pode ser definido como uma tempestade de idéias (SOMMERVILLE, 2001).

3.3.3 Modelagem e Documentação de Requisitos

3.3.3.1 Conceito

A etapa de modelagem de requisitos consiste na documentação formal, pelo uso de alguma ferramenta, dos dados coletados na elicitação de requisitos. A modelagem seleciona propriedades do domínio de aplicação que são do interesse do cliente (ISO/IEC, 2004).

O objetivo é abstrair o requisito de maneira que se possam identificar todos os detalhes bem como suas dependências com relação a outros requisitos. Os modelos utilizados para representação devem ter uma semântica bem definida. Os modelos de requisitos facilitam a comunicação entre os desenvolvedores do projeto de software (SAYAO, 2004).

Os requisitos podem ser modelados utilizando-se diversas técnicas, como DFDs (Diagramas de Fluxo de Dados), Tabelas de Decisão ou a modelagem de negócios utilizando o padrão IDEF0. Neste trabalho é enfatizada a modelagem utilizando-se casos de uso baseados em cenários, devido à facilidade de entendimento (são escritos na linguagem do problema), uma vez que este tipo de técnica ajuda a unificar critérios, na identificação de requisitos e no rastreamento dos requisitos (HAZAN, 2004).

3.3.3.2 Técnicas

3.3.3.2.1 Casos de Uso baseados em cenários

Júlio Leite define o cenário como uma descrição técnica e detalhada dos eventos, que permita a compreensão de seu funcionamento, elementos de informação e humanos requeridos, e objetivo pretendido. Segundo ele, é importante ter uma visão crítica do sistema em definição: não basta estruturar o ambiente e definir o cenário com as informações colhidas, é preciso, depois, analisá-lo ponderadamente, de forma crítica, questionando a validade de cada parte da estrutura e seu funcionamento, estabelecendo prioridades e eliminando procedimentos redundantes ou dispersivos e passos sem propósito, sem aplicação produtiva no processo interno da empresa (LEITE, 2001).

Um caso de uso pode ser entendido como um cenário que descreve como o software deve ser usado numa determinada situação. À medida que os requisitos são elicitados é possível criar um conjunto de cenários que identifique uma linha de uso para o sistema a ser construído. Para criar um caso de uso, o analista deve, primeiramente, identificar os diferentes atores, que

podem ser pessoas ou dispositivos, que usarão o sistema (PRESSMAN, 2002). Outro aspecto importante é que o caso de uso realiza um aspecto da funcionalidade (requisito) do software e pode representar roteiros de iteração com o usuário ou roteiros do manual de usuário, devendo gerar um ou mais benefícios para o cliente ou para os usuários (SAYAO, 2004). OS casos de uso podem ser aplicados para captar o comportamento pretendido pelo ator do sistema que está sendo desenvolvido, sem que seja necessário especificar como este comportamento será implementado (BOOCH, 2000).

Na figura 3, pode ser visto um exemplo de diagrama de uso de caso, proposto por Claudia Hazan (2004).

Caso de uso: Comprar ações na WEB
Escopo: Pacote financeiro(PAF)
Nível: objetivo do usuário
Interessados e interesses: (identificação dos atores)
Comprador – quer comprar ações e adicioná-las ao portifólio PAF automaticamente
Financeira – quer as informações de compra
Pré-condição: o usuário tem o PAF aberto
Pós-condições: site remoto tem conhecimento da compra, dos logs e o portifólio do usuário é atualizado
Cenário de sucesso – Principal:
1. Comprador seleciona ações na internet;
2. PAF pega o nome do web site a ser utilizado;
3. PAF abre conexão para o web site, retendo o controle;
4. Comprador navega e compra ações no site;
5. PAF intercepta respostas do site da web e atualiza portifólio;
6. PAF mostra o novo portifólio.

Exceções:
2a. Comprador seleciona um site que o PAF não trabalha
2a1. Sistema recebe novas sugestões do comprador, com opção de cancelar o caso de uso.

Figura 5 - Exemplo de caso de uso.
Fonte: HAZAN, 2004

3.3.3.3 Documentação de Requisitos

Os requisitos aprovados são documentados em um nível apropriado de detalhamento. Em geral, é utilizado um documento, chamado Especificação de Requisitos do Sistema, para que eles possam ser claramente compreendidos por todos os *stakeholders* (KOTONYA E SOMMERVILLE, 1998).

Desta forma, recomenda-se que os requisitos sejam documentados em uma linguagem natural e acompanhados de diagramas, para um melhor entendimento. Este documento deve retratar o

conhecimento do problema, em conformidade com as necessidades dos *stakeholders* e aos padrões de qualidade, relacionado ao que será produzido com tecnologia da informação para solução de problemas ou como oportunidade de negócio (ZANLORENCI, 2002).

O documento de requisitos é um documento formal utilizado para comunicar os requisitos aos clientes, e demais envolvidos no projeto. Ele descreve os serviços e funções que o sistema deve fornecer, as limitações sobre as quais o sistema deve operar; as definições de outros sistemas com o qual o sistema deve se integrar; as informações sobre o domínio da aplicação do sistema; as limitações nos processos usados para desenvolver o sistema; as descrições sobre o hardware no qual o sistema irá executar. O documento de requisitos deverá conter um capítulo introdutório que forneça um resumo do sistema, necessidades de negócio suportadas pelo sistema e um glossário que explica a terminologia utilizada (SOMMERVILLE, 2001).

Os requisitos são geralmente escritos em linguagem natural, complementados por diagramas e equações. Devem ser escritos de forma clara, pois diferentes pessoas podem entender os textos de diferentes formas. Não se deve omitir informações consideradas “óbvias”, pois o que é óbvio para alguns pode não ser óbvio para outros. Os diagramas e equações são importantes porque facilitam o entendimento (KOTONYA E SOMMERVILLE, 1998).

Existem diversas abordagens para escrever uma Especificação de Requisitos de Software. O Padrão IEEE/ANSI 830-1993 (IEEE, 1993) descreve uma estrutura genérica para o documento de Especificação de Requisitos de Software (ERS). O padrão do IEEE é genérico e pretende ser aplicado em uma variada gama de documentos de requisitos. Em geral, nem todas as partes do documento são necessárias para todos os documentos de requisitos. Cada organização deverá adaptar o padrão de acordo com suas necessidades.

Como exemplo, é mostrado, na figura 4 uma estrutura para o documento Especificação de Requisitos de Software, derivada do padrão da IEEE, sugerida por Peters (2001):

Índice analítico
1 Introdução
1.1 Objetivo (intenções e público alvo do documento)
1.2 Escopo (produtos de software a serem produzidos, capacidades, aplicações, benefícios relevantes, objetivos e requisitos do sistema)
1.3 Definições, acrônimos e abreviaturas
1.4 Referências (lista completa e, de preferência anotada, dos documentos referenciados na ERS)
2 Descrição geral
2.1 Perspectiva do produto
2.2 Funções do produto
2.3 Características do usuário
2.4 Restrições (políticas regulamentares, limitações de hardware, interfaces com outros aplicativos, operação paralela, funções de auditoria, funções de controle, requisito de linguagem de alto nível, protocolos de handshake de sinal, requisitos de confiabilidade, criticabilidade do aplicativo, segurança e perfil de segurança)
3 Especificação dos Requisitos
3.2 Requisitos funcionais
3.3 Requisitos não-funcionais
3.4 Critérios de aceitação dos requisitos
4 Rastreabilidade dos requisitos (artefatos de rastreabilidade)
Apêndices
Índice remissivo

Figura 6 - Padrão do documento de Especificação dos Requisitos do Sistema.
Fonte: Peters, 2001.

Na figura 5, é mostrado um exemplo que pode ser utilizado como parte do Documento de Especificação de Requisitos, no capítulo referente ao detalhamento dos requisitos específicos:

Detalhamento do Requisito

Identificação do requisito

Responsáveis pelo requisito

Nome	Função no Projeto

Solicitante

Nome	Função no Projeto

Data de solicitação

Artefatos relacionados

Tipo	Artefato	Comentário
	<link para o artefato>	

Origem

Prioridade

Riscos e impactos envolvidos

Risco	Impacto

Descrição detalhada

Recursos necessários

Tipo	Descrição	Comentários

Interfaces

Tipo	Descrição	Comentários

Layout sugerido

Objetos relacionados

Tipo	Objeto	Descrição	Comentário

Entradas e Saídas

Tipo	Descrição	Comentários

Figura 7 - Detalhamento do requisito.
Fonte: SAYÃO, 2004.

3.3.4 Análise de Requisitos

3.3.4.1 Conceito

Nesta etapa, deve haver uma cuidadosa avaliação dos requisitos, com ênfase em sua consistência e completude, ou seja, deve-se identificar possíveis problemas nos requisitos antes que o documento produzido sirva de base para o desenvolvimento do sistema. Assim, o objetivo da análise é a obtenção de uma especificação dos requisitos que seja consistente e completa. Na análise, efetuam-se duas tarefas, a verificação, que consiste em verificar se o produto está sendo construído da maneira correta em relação a outros produtos, e a validação, que tem por objetivo verificar se o produto está correto com relação à necessidade e intenção do cliente (SAYÃO, 2004).

A verificação é feita sobre os artefatos modelados na etapa anterior, e procura encontrar inconsistências e erros em sua especificação. Nesta atividade, os produtos de trabalho criados como conseqüência da elicitação e modelagem de requisitos devem ser verificados quanto à qualidade. A verificação procura responder a seguinte questão "estamos construindo o produto da maneira certa?"

A validação examina a especificação dos requisitos para garantir que todos foram estruturados de maneira não ambígua e estão de acordo com escopo do sistema a ser desenvolvido. A validação procura responder o seguinte questionamento "estamos construindo o produto certo em relação à intenção dos clientes?"

3.3.4.2 Técnicas de verificação

Criada em 1972 por Fagan, na IBM, a inspeção baseada em *checklist* é uma técnica de verificação que ainda é utilizada de forma ampla. Inspeções baseadas em *checklist* ajudam a encontrar de 30 a 90% dos erros antes que se passe a etapa seguinte (LEITE, 2002). Uma verificação pode ser feita efetuando uma leitura crítica do documento de requisitos contra um *checklist* de itens a serem verificados. O *checklist* é específico a cada tipo de artefato gerado, de modo que se tenha um *checklist* para os casos de uso, um para o documento de requisitos e assim por diante até atingir todos os artefatos serem inspecionados. Após a verificação, as inconsistências nos requisitos são ajustadas de maneira que é possível que seja necessário refazer as etapas de modelagem e até a elicitação.

Na tabela 1, é mostrado um exemplo de itens a serem verificados, usando-se um *checklist,* para verificação de requisitos. Observar que os critérios verificam questões de qualidade na especificação dos requisitos e podem ser aplicados a todos os artefatos envolvidos na especificação dos requisitos (casos de uso, glossários e o documento de requisitos). Outra forma de efetuar a verificação é utilizar um *checklist* para cada tipo de artefato. Para fins didáticos, o exemplo será dividido em seções, identificando as questões pertinentes a cada artefato.

***Checklist* de Verificação**			
Questões – Casos de uso	**OK**	**NOK**	**Observações**
O caso de uso representa uma tarefa específica, com um único objetivo, tendo início, meio e fim?	☐	☐	
O objetivo do caso de uso está claro?	☐	☐	
Os atores envolvidos com o caso de uso estão bem definidos?	☐	☐	
O caso de uso está livre de detalhes de implementação e projeto?	☐	☐	
O caso de uso está com seus fluxos alternativos bem documentados?	☐	☐	
O caso de uso está com seus fluxos de exceção bem documentados?	☐	☐	
A seqüência do diálogo para cada fluxo está escrita de forma clara, sem ambigüidade?	☐	☐	
As pré e pós-condições estão presentes no caso de uso?	☐	☐	
Todas as regras de negócio referenciadas no caso de uso estão detalhadas?	☐	☐	
Questões – Requisitos	**OK**	**NOK**	
Clareza - Os requisitos estão claros e bem determinados?	☐	☐	
Completude – Estão presentes todos os requisitos necessários à especificação do sistema?	☐	☐	
Consistência – Os requisitos estão consistentes com a visão geral do sistema?	☐	☐	
Corretude – Os requisitos descrevem as funcionalidades de maneira correta?	☐	☐	
Funcionalidades – As funcionalidades descritas são suficientes para atingir os objetivos do sistema?	☐	☐	
Testabilidade – As funcionalidades permitem testes de forma a mostrar que os requisitos foram satisfeitos?	☐	☐	
Detalhamento – O nível de detalhamento é suficiente para fornecer base ao design do sistema?	☐	☐	
Questões – OO	**OK**	**NOK**	
Todas as classes são representadas por retângulos com 1,2 ou 3 compartimentos?	☐	☐	
As classes possuem nomes diferentes?	☐	☐	
Existem classes sem relacionamentos definidos?	☐	☐	
Os atributos e os métodos de cada classe são adequados aquela classe?	☐	☐	
Todo comportamento é possível de ser contemplado através das associações do modelo?	☐	☐	

Tabela 3 - Exemplo de *checklist* de verificação dos requisitos.
Fonte: Sayão, 2004.

Dependendo do foco da verificação, mais itens podem ser acrescentados, em acordo com a metodologia e padrões de qualidade da organização ou específicos ao tipo de sistema a ser desenvolvido.

3.3.4.3 Técnicas de validação

Para KOTONYA e SOMMERVILLE (1998), o mecanismo primário de validação de requisitos é a revisão técnica. O time de revisão inclui os engenheiros e desenvolvedores do sistema, clientes, usuários e outros *stakeholders,* que examinam os requisitos do sistema à procura de erros de conteúdo ou interpretação, pontos onde pode ser necessário esclarecimento, perda de informações, inconsistências, requisitos que são conflitantes com outros, ou requisitos irreais (de desenvolvimento impossível).

Embora a revisão para validação dos requisitos possa ser conduzida de qualquer maneira desde que possibilite a descoberta de erros nos requisitos, é útil examinar cada requisito contra um *checklist* (HAZAN, 2004). Na tabela 2 é sugerido um pequeno subconjunto de critérios do que pode ser questionado:

***Checklist* de Validação**			
Questão	**OK**	**NOK**	**Observações**
Os requisitos estão estruturados claramente?	☐	☐	
Não há problemas de interpretação incorreta?	☐	☐	
A fonte (pessoa, documento etc.) foi identificada?	☐	☐	
A estrutura final do requisito foi examinada pela ou contra a fonte original?	☐	☐	
O requisito viola alguma regra de negócio?	☐	☐	
O requisito é passível de teste?	☐	☐	
Se sim, podem ser especificados os testes para exercitar os requisitos?	☐	☐	
O requisito é rastreável para qualquer modelo de sistema que seja criado?	☐	☐	
O requisito é rastreável para os objetivos gerais do produto/sistema?	☐	☐	
Os requisitos associados a desempenho, comportamento e características operacionais foram estruturados claramente?	☐	☐	

Tabela 4 - Exemplo de *checklist* de validação dos requisitos.
Fonte: Hazan, 2004.

Da mesma forma, como ferramenta auxiliar para a validação, pode-se utilizar protótipos. A utilização de protótipos ajuda a validar os atores, explicando o que acontece a eles e

descrevendo como acontece. Os protótipos são mais eficazes para projetos que tiverem um conteúdo desconhecido ou inovador. Além disso, os protótipos são baratos, fáceis de criar ou modificar e fornecem uma crítica das interfaces do sistema cedo no desenvolvimento (SAYÃO, 2004).

3.4 CONCLUSÕES DO CAPÍTULO

Neste capítulo foram mostrados os aspectos importantes da disciplina de Engenharia de Requisitos, tais como os processos de elicitação, modelagem, documentação e análise de requisitos, bem como exemplos de técnicas que podem ser utilizadas para a sua execução. É importante conhecer o processo de levantamento de requisitos, pois durante o gerenciamento é possível que os requisitos levantados inicialmente sofram alterações, sejam excluídos, ou ainda, novos requisitos podem surgir, o que pode necessitar modelagem levantamento destes.

Paralelamente às atividades de levantamento de requisitos, deve-se desenvolver o gerenciamento de requisitos. Essa atividade consiste em administrar as inevitáveis mudanças dos requisitos propostos. Tais mudanças surgem, principalmente, quando são alteradas as prioridades do negócio, quando se identificam erros ou omissões nos requisitos ou quando novos requisitos são definidos. Gerenciamento e rastreamento de requisitos são reconhecidos como importantes pré-requisitos para desenvolver software de alta qualidade (WIEGERS, 1999). Em função do escopo deste trabalho, será enfatizado o gerenciamento de requisitos de forma que este processo será discutido em detalhes no próximo capítulo.

Desta forma, as informações vistas neste capítulo, servirão de subsídio para a avaliação dos requisitos que é uma das etapas iniciais abordadas pela área de processo de gerência de requisitos do CMMI.

4 GERÊNCIA DE REQUISITOS

4.1 INTRODUÇÃO

Durante os estágios do ciclo de vida do produto de software, novos requisitos surgem e requisitos existentes são modificados. Estas mudanças dos requisitos durante o processo de desenvolvimento de software fazem parte deste processo. Dentre os motivos para que ocorram alterações encontram-se necessidades não identificadas inicialmente, alterações no contexto onde o sistema está inserido, correção de erros detectados por processos de qualidade ou mesmo novas perspectivas por parte dos interessados. Em muitos casos mais de 50% dos requisitos do sistema sofrerão modificações antes de terem sido completados e, obviamente, isto pode causar sérios problemas para os envolvidos no desenvolvimento e gerenciamento do projeto (KOTONYA E SOMMERVILLE, 1998).

Assim, todas as mudanças devem ser acompanhadas, de forma a garantir que todos os artefatos afetados por uma alteração de requisitos sejam alterados. As atividades ligadas ao processo de acompanhamento de mudanças são denominadas de Gerência de Requisitos (LEITE, 2005).

Desta forma, o objetivo da gerência de requisitos é garantir que as solicitações dos clientes sejam controladas durante processo de produção de software. Estas solicitações devem estar relacionadas ao produto de software que é descritos pelo conjunto de requisitos sob gerência.

Para isso, é importante que se defina, claramente, o escopo (a combinação de funcionalidade, recursos e tempo) para o sistema. Pode-se afirmar também que a gerência de requisitos procura gerenciar o escopo, controlar os pedidos de alterações, manter uma rastreabilidade (relação entre os requisitos) e gerenciar as inconsistências entre o documento de requisitos e demais artefatos produzidos durante o processo de construção do software (HAZAN, 2004). A relação entre estas atividades pode ser vista na figura 6.

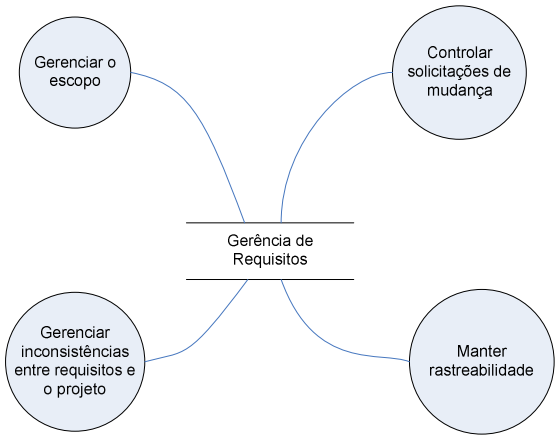


Figura 8 - Atividades da gerência de requisitos.
Fonte: Sayão, 2004

Como o escopo deste trabalho busca o alinhamento da atividade de gerência de requisitos com o modelo CMMI, a próxima seção mostrará uma visão geral do modelo CMMI, identificando onde a gerência de requisitos é tratada. Após esta etapa, serão discutidas as atividades da gerência de requisitos, buscando mostrar como estas atividades estão relacionadas com o modelo.

4.2 PROCESSO DE GERÊNCIA DE REQUISITOS

O processo de gerência dos requisitos define e mantém os requisitos durante o ciclo de vida de um sistema. A entrada deste processo pode ser um novo requisito, já definido pelo processo de definição de requisitos, ou uma mudança em um requisito existente. Fazem parte deste processo as atividades de entendimento do novo requisito ou da mudança, a análise de viabilidade e impacto e a realização das mudanças, quando aprovadas. Desta forma, a gerência de requisitos inclui o planejamento e a gerência das mudanças sobre os requisitos de um sistema (LEFFWINGGELL, 2003).

As atividades de gerência de requisitos são atendidas, no modelo CMMI, pela área de processo de Gerência de Requisitos. Segundo o modelo, a Gerência de Requisitos visa estabelecer um entendimento comum entre o cliente e o fornecedor quanto aos requisitos que serão atendidos pelo projeto de software. Esta área de processo descreve as atividades para obter e controlar as mudanças de requisitos e assegurar a atualização dos planos do projeto e demais artefatos que sejam atingidos por uma mudança nos requisitos.

Desta forma, gerenciar requisitos é realizar uma seqüência de eventos dinâmicos e freqüentemente recursivos, sendo os mesmos fundamentais para um processo de engenharia

controlado e disciplinado. Além disso, este processo estabelece e mantém um acordo formal entre clientes/usuários e a equipe do projeto sobre os requisitos e suas mudanças ao longo do projeto (SEI, 2005).

O objetivo específico do CMMI-SW, versão 1.1, para a Área de processo de Gerência de Requisitos é (SEI, 2002):

"*Requisitos são gerenciados e as inconsistências, com o planejamento do projeto e o produto, são identificadas*."

O objetivo principal de qualquer produto é alcançar as necessidades dos *stakeholders* e dos clientes. Mas isso passa por regras internas de funcionalidade do software e pela necessidade de qualidade no processo. A área de processo de Gerência de Requisitos cria um ciclo vital para facilitar este processo.

Esta área de processo não abrange diretamente a definição dos requisitos, pois se presume que estes já existem inicialmente, e serão gerenciados. Mas, como as primeiras práticas específicas desta área, indicadas no modelo, prevêem uma avaliação destes requisitos, eles poderão ser alterados durante este processo de gerência (SEI, 2002).

As práticas que precisam ser estabelecidas para atender os objetivos da Gerência de Requisitos do CMMI são listadas abaixo:

- A organização precisa definir o conjunto de requisitos a serem alcançados (SP-1.1).
- A organização precisa garantir que os requisitos são alcançados (SP-1.2).
- A organização precisa gerenciar as mudanças nestes requisitos (SP-1.3 e SP-1.4).
- A organização precisa garantir que o planejamento do projeto, os produtos e requisitos são consistentes (SP-1.5).

Desta forma, a gerência de requisitos procura garantir que o escopo do projeto está definido e sob controle, que tanto o solicitante, quanto a equipe do projeto, estão de acordo que aquele é o escopo do projeto e que, quando os requisitos mudarem, o impacto será avaliado e o projeto será re-planejado, se necessário.

A seguir serão discutidas as atividades da gerência de requisitos, mostrando suas práticas e seus relacionamentos. Em função do escopo deste trabalho, estas atividades estarão alinhadas às práticas sugeridas pelo CMMI na área de processo de gerência de requisitos. Desta forma,

serão discutidas as atividades da gerência de requisitos e a sua relação com as praticas definidas pelo modelo.

4.3 CONTROLE DE ALTERAÇÕES

A gerência de alterações dos requisitos envolve processos, procedimentos e padrões que são usados para gerenciar as alterações dos requisitos do sistema (KOTONYA e SOMMERVILLE 1998). Este gerência garante que sejam coletadas todas as informações relacionadas aos envolvidos na alteração, além de ser realizada, para cada alteração proposta, uma avaliação de custos e benefícios. Sem gerência de alteração formal é impossível garantir que as alterações propostas aos requisitos têm suporte nas metas de negócio fundamentais do sistema. Para garantir uma abordagem consistente à gerência de alteração, a organização deve definir um conjunto de práticas de gerência de alterações que cubra:

- O processo de solicitação de alteração e a informação requerida para processar cada solicitação de alteração;
- O processo usado para analisar o impacto e custos da alteração e informações de rastreabilidade associadas;
- O grupo que considera formalmente as solicitações de alteração. Este grupo é independente para fazer as considerações sobre as solicitações de alteração para que possam ser tomadas decisões objetivas sobre a contribuição da mudança com relação às metas gerais do sistema e reais necessidades da alteração.

Uma alteração engloba da mudança em um requisito existente, a exclusão de um requisito ou a inclusão de um novo requisito no sistema ou projeto. Durante a atividade de controle de mudanças, algumas etapas são desenvolvidas sempre que uma solicitação de mudança ocorre: a solicitação de mudança é analisada, o impacto da sua aceitação são discutidos e as soluções são propostas; os custos e o impacto da mudança em outros requisitos são avaliados; os documentos de requisitos e demais artefatos afetados são alterados para refletir a mudança solicitada.

O processo de mudança inicia por uma solicitação de mudança de requisitos. Esta solicitação pode vir de qualquer *stakeholder* do projeto. Uma solicitação de mudança de requisito pode refletir alguma diferença entre o que foi planejado e o que está sendo executado no projeto.

Estas diferenças podem incluir alterações em qualquer requisito do sistema (SAYÃO, 2004). Na figura 9 é mostrado um exemplo simplificado de formulário para solicitação de mudança de requisitos.

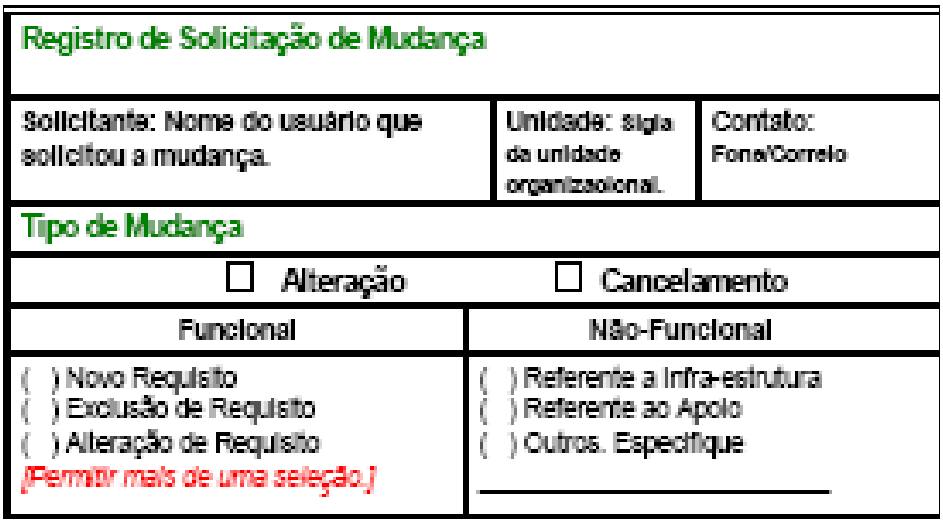


Figura 9 - Modelo do documento simplificado de solicitação de alteração de requisitos.
Fonte: Sayão, 2004.

Após o recebimento da solicitação, é feita uma revisão desta de forma a obter um entendimento comum, entre os envolvidos, sobre a mudança que está sendo solicitada. Para evitar que os requisitos cresçam indistintamente, são estabelecidos critérios para determinar quais são os fornecedores apropriados a partir dos quais se deve receber estes requisitos, utilizando-se das solicitações de mudança. As atividades de recebimento dos requisitos executam uma verificação, junto ao fornecedor, sobre os requisitos recebidos para assegurar que um entendimento compartilhado foi conseguido. O resultado desta análise e diálogo é um conjunto de requisitos acordados.

Desta forma, é fundamental estabelecer critérios para designar fornecedores apropriados dos quais são recebidas as solicitações de mudança e, da mesma forma, conduzir análise dos requisitos com o fornecedor de requisitos para garantir um entendimento compatível e compartilhado do significado dos requisitos.

O modelo CMMI indica os seguintes artefatos que são considerados típicos como resultado da documentação da atividade de revisão da solicitação de mudança de requisitos (SEI, 2002):

- Lista de Critérios para identificar fornecedores de requisitos apropriados
- Lista de critérios para utilizados para avaliação e aceite de requisitos
- Resultados de análise dos requisitos, utilizando os critérios estabelecidos

- Definição de um conjunto de requisitos acordados

A próxima etapa é identificar o esforço, prazo e custo que será necessário para executar as mudanças solicitadas. Assim, uma vez tendo um entendimento do que deverá ser alterado, a equipe técnica efetua uma análise para definir qual será o esforço real para implementar as alterações. Para executar esta análise pode ser necessário retornar à etapa da análise e validação do requisito, utilizando as técnicas mostradas no capítulo 2 (SAYÃO, 2004). Para auxiliar a análise de impacto, sugere-se a utilização de uma matriz de rastreabilidade entre os requisitos e entre os requisitos e os demais artefatos do projeto.

Uma matriz de rastreabilidade mostra as dependências entre um requisito e os demais requisitos e artefatos do projeto. Desta forma é possível identificar o quê será afetado, no sistema, pela mudança em um requisito e verificar se a alteração no esforço afetará significativamente os prazos e custos do projeto. Na sessão 3.5, deste capítulo, será feito um estudo sobre rastreabilidade, suas técnicas e artefatos.

Com o resultado desta análise, é feito um estudo de impacto para que se possa definir se a solicitação de mudança do requisito será executada ou não.

Se for definido que a solicitação de mudança de requisito não será aceita, a próxima etapa será comunicar, formalmente os *stakeholders* e o solicitante e registrar esta negação da solicitação para que se tenha um histórico das solicitações. Este histórico servirá de base para as decisões das próximas solicitações de mudança que serão recebidas. Da mesma forma, o modelo CMMI prevê que é necessário registrar a razão pelo qual uma mudança solicitada não tenha sido aceita.

Caso seja definido que a solicitação de mudança seja implementada no projeto, será necessário identificar as inconsistências entre os artefatos do projeto e os requisitos alterados, excluídos ou incluídos pela solicitação.

Uma vez identificadas estas inconsistências, elas devem ser registradas e, em seguida, devem iniciar as respectivas ações corretivas para adequar os documentos, atividades e demais artefatos à mudança aceita. Desta forma, a mudança solicitada será executada no projeto e refletirá na alteração do cronograma e na lista de atividades (BACCA, 2005). Para a execução desta atividade, o modelo CMMI sugere algumas atividades: revisão dos planos do projeto, e demais artefatos identificados como afetados pela análise de impacto da mudança.

Esta atividade de identificação de inconsistências entre os requisitos e demais artefatos do projeto não é executada somente durante o controle de mudanças, durante todo o ciclo de vida do projeto como uma atividade contínua (REINEHR, 2006).

Para melhor entendimento, as etapas da atividade de controle de mudanças são mostradas na figura 10.

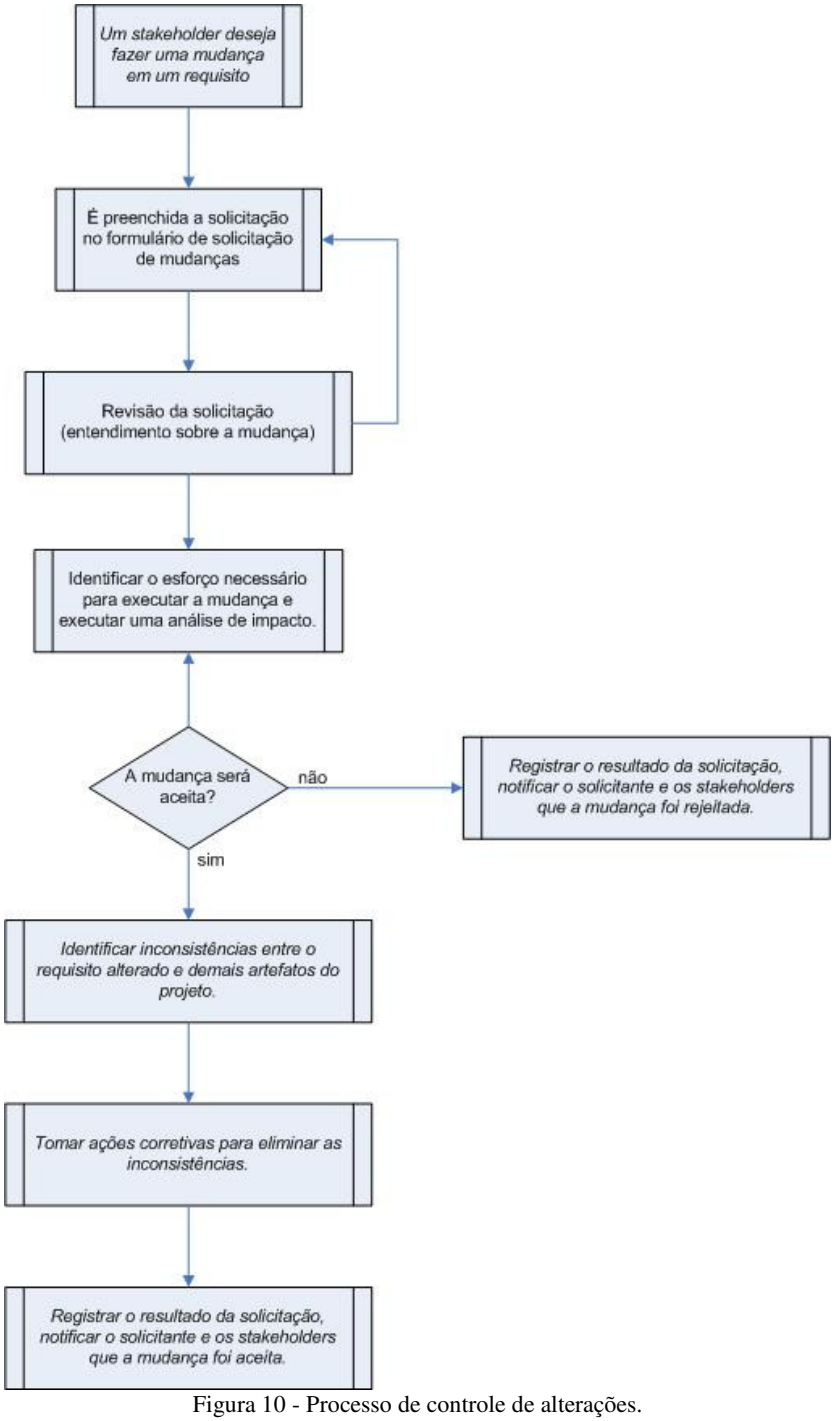

Um stakeholder deseja fazer uma mudança em um requisito
É preenchida a solicitação no formulário de solicitação de mudanças
Revisão da solicitação (entendimento sobre a mudança)
Identificar o esforço necessário para executar a mudança e executar uma análise de impacto.
A mudança será aceita?
não
Registrar o resultado da solicitação, notificar o solicitante e os stakeholders que a mudança foi rejeitada.
sim
Identificar inconsistências entre o requisito alterado e demais artefatos do projeto.
Tomar ações corretivas para eliminar as inconsistências.
Registrar o resultado da solicitação, notificar o solicitante e os stakeholders que a mudança foi aceita.

Figura 10 - Processo de controle de alterações.
Fonte: BACCA, 2005

Para dar suporte ao processo de controle de mudança, Claudia Bacca (2005), sugere um modelo de documento de solicitação de mudança mais completo, que poderá ser utilizado em todas as etapas discutidas anteriormente. Este modelo inclui campos para suportar deste a solicitação, o entendimento, a análise da mudança até sua aprovação ou não pelo gerente do projeto. Este modelo pode ser visto na figura 11.

Formulário de Solicitação de Mudanças			**Número:**
Sessão 1: Preenchida pelo Solicitante			
Projeto:	*Nome do projeto*	**Data:**	*Data da solicitação*
Solicitante:	*Nome do solicitante*		
Gerente do Projeto:	*Nome do gerente do projeto*		
Descrição da solicitação:			
Descrição da mudança desejada			
Motivo da solicitação:			
Motivo da solicitação			
Riscos se a solicitação não for implementada:			
Descrição dos possíveis riscos se a mudança não for implementada			
Sessão 2: Preenchida pelo Gerente do Projeto			
Esforço adicional necessário:	*Esforço adicional para implementação*	**Duração adicional necessária:**	*Duração adicional para implementação*
Recursos adicionais necessários:	*Recursos adicionais necessários para implementação da mudança*	**Dependências:**	*Requisitos e demais artefatos afetados pela implementação da mudança*
Custo adicional necessário	*Custo adicional necessário para implementação da mudança*		
Riscos se implementado:	*Riscos se a mudança for implementada*		
Recomendações do Gerente do Projeto			
Recomendações do Gerente do Projeto referente à avaliação da solicitação da mudança.			
Sessão 3: Preenchida pela gerência de nível superior/Patrocinador do Projeto			
Recomendações do Patrocinador do Projeto			
Recomendações do Patrocinador do Projeto referente à avaliação da solicitação da mudança.			
Assinatura do Patrocinador do Projeto:		**Data:**	
Disposição	[ ] aprovada [ ] rejeitada	**Incluído na versão/data:**	

Figura 11 - *Template* de um exemplo de Solicitação de Mudanças de Requisitos.
Fonte: BACCA, 2005

4.4 RASTREABILIDADE

Uma parte crítica da gerência de alterações é a avaliação do impacto da mudança sobre resto do projeto. Se a mudança é proposta enquanto os requisitos estão sendo desenvolvidos, deve ser identificado como esta alteração afeta outros requisitos. Se a alteração é proposta enquanto o sistema está em implementação, o impacto de alteração envolve verificar como esta alteração afeta os requisitos, o design do sistema e sua implementação. Esta avaliação de impacto ocorrerá independente do ciclo de vida adotado para o desenvolvimento da aplicação. Se a alteração é proposta depois que o sistema foi colocado em operação, deve haver também uma verificação adicional a fim de identificar como todos os *stakeholders* do sistema podem ser afetados pela alteração (KOTONYA E SOMMERVILLE, 1998).

O modelo CMMI sugere que quando os requisitos são gerenciados, a rastreabilidade pode ser estabelecida, desde um requisito fonte até seus requisitos de mais baixo nível e destes de volta para seu requisito fonte. A rastreabilidade pode cobrir os relacionamentos com outros artefatos, como tarefas de trabalho, planos de testes, mudanças na documentação de design etc, e é particularmente necessária na condução da análise de impacto das mudanças de requisitos nos planos do projeto, atividades e demais artefatos.

Para conduzir este tipo de avaliação de impacto, informações sobre dependências de requisitos, razão do requisito e sua implementação devem ser mantidas para complementar as informações do documento de requisitos. Estas informações são chamadas usualmente de informações de rastreabilidade.

Desta forma, o rastreamento de requisitos é utilizado para prover relacionamentos entre requisitos, arquitetura e implementação final do sistema e possibilita uma adequada compreensão dos relacionamentos de dependência entre requisitos e entre os demais artefatos gerados para o sistema. A rastreabilidade pode ser implementada por um conjunto de associações ou ligações (*links*) entre requisitos inter-relacionados, entre requisitos e suas fontes, e entre requisitos e os componentes que os implementam (LEITE, 2005). As avaliações de impacto de alteração dependem das informações de rastreabilidade para encontrar os requisitos afetados por uma proposta de alteração.

Segundo KOTONYA e SOMMERVILLE (1998), as informações de rastreabilidade são classificadas em quatro tipos:

- Rastreabilidade “de – para trás”: *links* dos requisitos a suas fontes (outros documentos ou pessoas);
- Rastreabilidade “de – para frente”: *links* dos requisitos a seus componentes de design e implementação;
- Rastreabilidade “volta – para trás”: *links* dos componentes de design e implementação de volta para os requisitos;
- Rastreabilidade “volta – para frente”: *links* de documentos (que precederam o documento de requisitos) aos requisitos relevantes.

A figura 12 mostra como as ligações (*links*) possibilitam acompanhar o ciclo de vida de um requisito, em ambas as direções.

Figura 12 - Rastreabilidade do requisito.
Fonte: Leite, 2005

No primeiro tipo temos a rastreabilidade *forward-to* (para frente), que liga documentos obtidos no processo de elicitação a requisitos relevantes, e a rastreabilidade *backward-from* (para trás), que liga requisitos às suas fontes. No segundo tipo temos rastreabilidade *forward-from*, que liga requisitos a artefatos de desenho e implementação e rastreabilidade *backward-to*, que liga artefatos de desenho e implementação de volta a requisitos (SAYAO, 2004).

Potencialmente, a rastreabilidade cobre um volume muito grande de informações, porém o gerente do projeto deve definir políticas de rastreabilidade identificando quais informações de rastreabilidade são essenciais e devem ser mantidas (KOTONYA E SOMMERVILLE, 1998).

Um artefato que é utilizado para visualizar de maneira ágil e rápida a ligação entre um requisito e os demais artefatos de um sistema é a matriz de rastreabilidade. A matriz de rastreabilidade oferece como grande facilitador a visualização global dos artefatos do sistema e requisitos do sistema em uma tabela de forma gráfica, dando suporte para tomar decisões, descobrir problemas e sua solução de forma mais rápida, muitas vezes antes da fase de implementação.

Além da matriz de rastreabilidade entre requisitos, é possível relacionar quaisquer outros artefatos do sistema. Um exemplo de matriz de rastreabilidade, entre casos de uso (UC) e requisitos funcionais (RF), é mostrado na figura 13.

	RF1-Cadastro de Usuários	RF2-Níveis hierárquicos	RF3-Cadastro de Clientess	RF4-Controle de Notas Fiscais	RF5-Cadastro de Produtos	RF6-Cadastro de Condições de Pagamento	RF7-Cálculo de Comissão	RF14-Emitir relatório de totais de vendas
UC1-Cadastrar usuários	☒	☐	☐	☐	☐	☐	☐	☒
UC2-Cadastrar clientes	☒	☐	☒	☐	☐	☐	☐	☐
UC3-Emitir Notas Fiscal	☐	☐	☒	☒	☐	☒	☐	☐
UC4-Calcular comissão	☐	☐	☐	☒	☐	☐	☐	☐

Figura 13 – Exemplo de Matriz de rastreabilidade entre requisitos e casos de uso.
Fonte: KOTONYA e SOMMERVILLE (1998)

Na figura 14, temos um exemplo de matriz de rastreabilidade entre casos de uso (UC) e requisitos funcionais (FEAT), obtida em um dos softwares que serão analisados, o Rational RequisitePro.

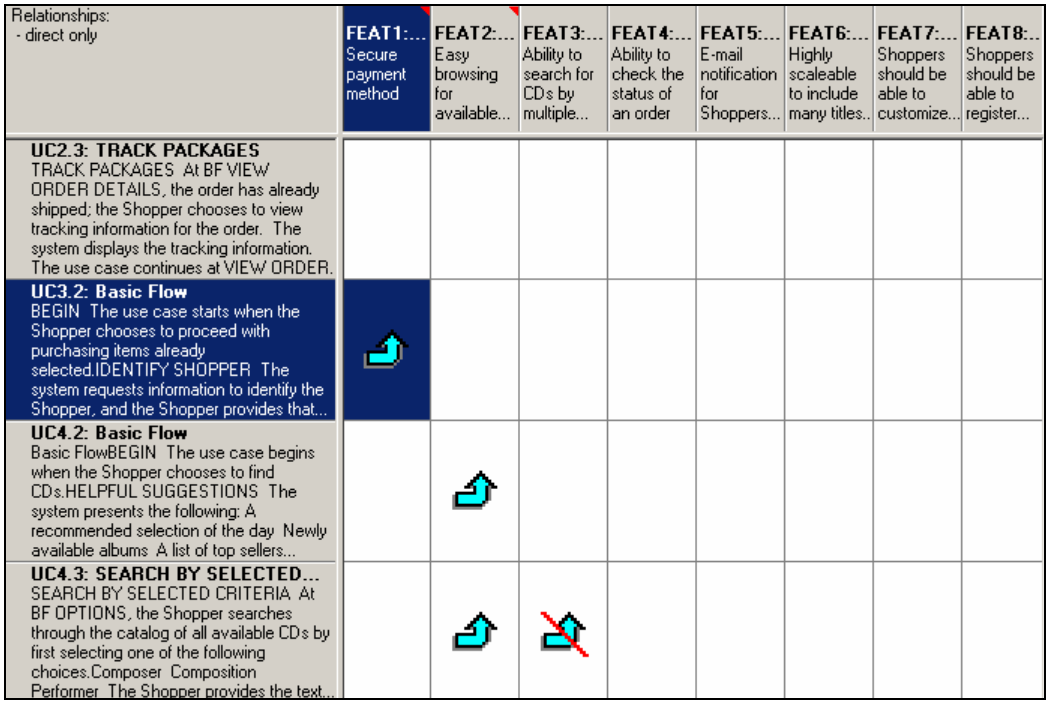

Relationships:
- direct only
FEAT1:... Secure payment method
FEAT2:... Easy browsing for available...
FEAT3:... Ability to search for CDs by multiple...
FEAT4:... Ability to check the status of an order
FEAT5:... E-mail notification for Shoppers...
FEAT6:... Highly scaleable to include many titles...
FEAT7:... Shoppers should be able to customize...
FEAT8:... Shoppers should be able to register...
UC2.3: TRACK PACKAGES
TRACK PACKAGES At BF VIEW ORDER DETAILS, the order has already shipped; the Shopper chooses to view tracking information for the order. The system displays the tracking information. The use case continues at VIEW ORDER.
UC3.2: Basic Flow
BEGIN The use case starts when the Shopper chooses to proceed with purchasing items already selected.IDENTIFY SHOPPER The system requests information to identify the Shopper, and the Shopper provides that...
UC4.2: Basic Flow
Basic FlowBEGIN The use case begins when the Shopper chooses to find CDs.HELPFUL SUGGESTIONS The system presents the following: A recommended selection of the day Newly available albums A list of top sellers...
UC4.3: SEARCH BY SELECTED...
SEARCH BY SELECTED CRITERIA At BF OPTIONS, the Shopper searches through the catalog of all available CDs by first selecting one of the following choices.Composer Composition Performer The Shopper provides the text...

Figura 14 - Matriz de rastreabilidade com indicação de inconsistência entre os artefatos.
Fonte: IBM Rational RequisitePro (2005)

Observa-se, neste exemplo, que a matriz de rastreabilidade já representa o tipo de rastreabilidade pela direção da seta (rastreabilidade para frente e para trás). Além disso, a matriz mostra, com um traço sobre a seta, os artefatos que foram alterados indicando a necessidade de verificação destas inconsistências.

Outra forma de mostrar a rastreabilidade entre artefatos é a visualização em forma de árvore de rastreabilidade. Da mesma forma, são mostrados os artefatos que "dependem" e são "dependentes de", porém na forma hierárquica. Na figura 15, pode-se observar um exemplo que rastreia, a partir de uma tarefa do projeto (identificada como MSPT) para um requisito funcional (identificadas por FEAT) e este para os casos de uso associados ou, ainda, os requisitos não-funcionais (identificados por SUPP).

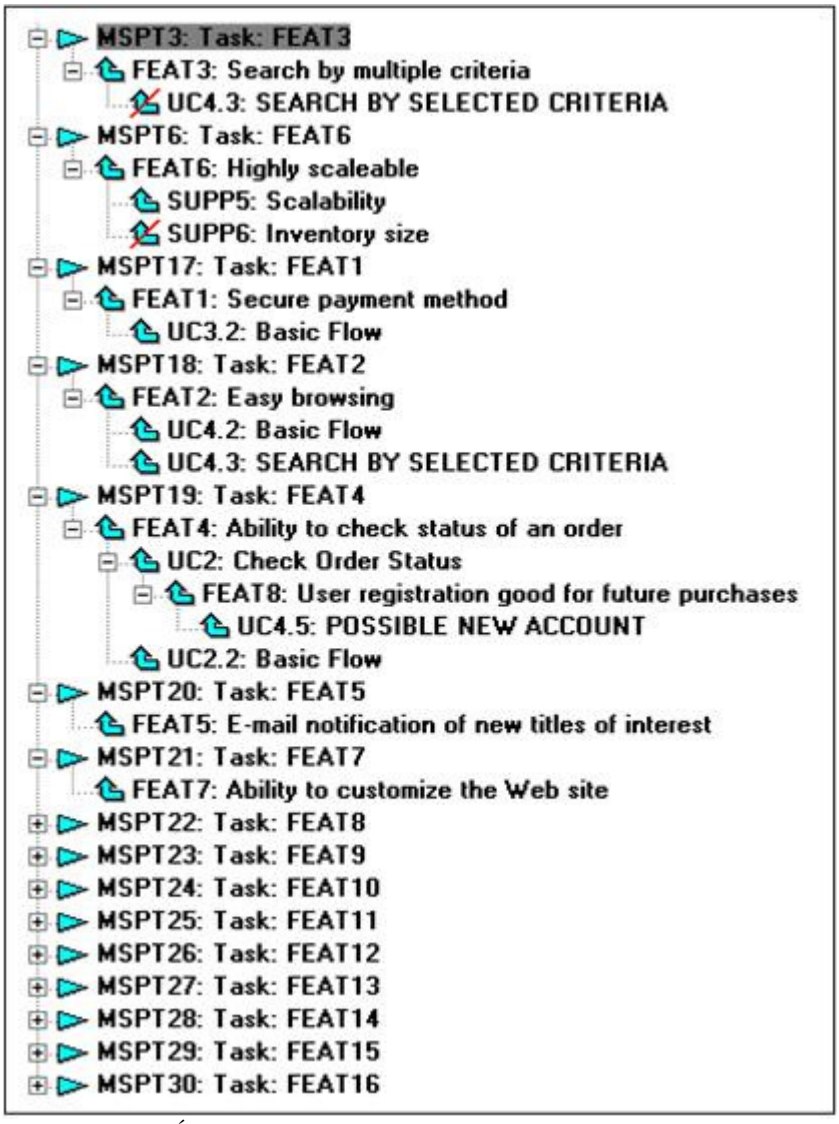


Figura 15 - Árvore de rastreabilidade ente requisitos e artefatos do sistema.
Fonte: Rational RequisitePro (2005)

Tanto a matriz de rastreabilidade como a árvore de rastreabilidade podem ser visualmente impraticáveis quando o número de artefatos relacionados é muito grande. Como alternativa, procura-se gerar visões da matriz ou árvore de forma a termos várias matrizes, cada uma mostrando a rastreabilidade específica entre tipos determinados de artefatos.

Outra forma de manter a rastreabilidade é a utilização de bancos de dados de requisitos, que mantenham informações de relacionamentos de dependência “muitos para muitos” entre os requisitos e demais artefatos do sistema. Desta forma, a navegação entre as informações de rastreabilidade é colocada, também, como uma solução que melhora a rastreabilidade.

4.5 CONCLUSÕES DO CAPÍTULO

Este capítulo abordou os o processo de gerência de requisitos trazendo alguns elementos da área de processo Gerência de Requisitos do modelo CMMI. Com isso conseguiu-se mostrar algumas técnicas para a realização das atividades de gerência de requisitos visando um alinhamento com as práticas definidas pelo modelo.

Desta forma, é possível ter uma visão mais clara de como o objetivo da área de Gerência de Requisitos pode ser alcançado utilizando-se as técnicas e artefatos mostrados.

O conhecimento destes conceitos servirá de subsidio para a avaliação das ferramentas de gerência de requisitos que será abordada no capítulo 4.

5 AVALIAÇÃO DAS FERRAMENTAS

5.1 INTRODUÇÃO

As empresas que buscam a melhoria de processo ou até mesmo a certificação no modelo CMMI procuram utilizar ferramentas de apoio que estejam alinhadas ao que é proposto pelo modelo. Na área de gerência de requisitos tem-se a mesma necessidade, ou seja, é importante que a ferramenta utilizada para o gerenciamento dos requisitos de software esteja em acordo com os objetivos propostos pelo CMMI.

Após a etapa de estudo das técnicas de Engenharia de Requisitos e características do modelo CMMI juntamente com a sua área de processo de gerência de requisitos será executada a avaliação das ferramentas de gerência de requisitos. A avaliação é feita usando como base no método de avaliação oficial da SEI, o SCAMPI (*Standard CMMI Appraisal Method for Process Improvement*). Como este método de avaliação foi desenvolvido para a execução de avaliações formais do processo das organizações, é necessário uma adaptação, em função do escopo deste trabalho, para a avaliação das ferramentas. Esta adaptação é descrita na próxima sessão, onde é descrito o processo de avaliação utilizado.

5.2 PROCESSO DE AVALIAÇÃO

O SCAMPI (*Standard CMMI Appraisal Method for Process Improvement*) é o método de avaliação oficial de processos do modelo CMMI (SEI, 2001). O método foi desenvolvido com o objetivo de avaliar os processos de uma organização relacionando estes processos com os objetivos do CMMI. Neste sentido, o SCAMPI permite à organização identificar os pontos fortes e os pontos fracos de seus processos, relacionando estes pontos com o modelo CMMI, priorizar seus planos de melhoria de processo focando nos pontos mais importantes. Da mesma forma, o SCAMPI classifica a organização, definindo os níveis de capacidade dos processos (se a organização estiver optando pela representação contínua) ou nível de maturidade (se a organização estiver optando pela representação em estágios).

O SCAMPI busca evidências que indiquem que as práticas propostas pelo modelo foram atingidas. Estas evidências são coletadas efetuando-se uma análise de documentos (artefatos), apresentações (material apresentado pela organização), instrumentos (informação escrita, como por exemplo, questionários, mapeamentos dos processos do CMMI para os processos da organização) e entrevistas. Neste sentido, esta avaliação buscará evidências, nas ferramentas avaliadas, que indiquem se a ferramenta implementa ou não as práticas propostas pela área de processo de Gerência de Requisitos. As evidências serão obtidas pela análise da ferramenta e da documentação, fornecida pelo fabricante.

No SCAMPI, os indicadores de implementação de uma prática são as evidências de implementação das práticas do CMMI. Por exemplo, a existência de um artefato pode ser oferecido como resultado direto da implementação de uma prática do modelo. Outros indicadores podem oferecer evidências indiretas da implementação de uma prática, como, por exemplo, a apresentação de *templates* para registro de reuniões de status de um projeto. Estas evidências são indiretas porque não garantem que o objetivo destas reuniões seja atingido, sendo necessário que se recomende a execução de determinada prática durante a reunião. Desta forma, o SCAMPI provê tipos de indicadores, que são mostrados na tabela 5.

Tipos de Indicadores de Implementação de uma prática		
Tipo	Descrição	Exemplos
Artefatos Diretos	As saídas tangíveis resultando diretamente da implementação de uma prática específica ou de uma prática genérica.	Produtos de trabalho típicos indicados pelas práticas do CMMI. Documentos, produtos entregues, materiais de treinamento etc.
Artefatos Indiretos	Artefatos que confirmam a implementação de uma prática específica ou genérica mas que não são gerados com objetivo direto de implementar a prática. Este indicador é útil quando se tem dúvidas se a prática foi realmente atingida (por exemplo, um artefato existe mas não há indicações de onde ele foi gerado, que se trabalhou no seu desenvolvimento ou como o artefato realmente é utilizado no processo).	Produtos de trabalho típicos indicados pelas práticas do CMMI. Modelos de documentos para Atas de reunião, relatórios de status e métricas de performance.

Tabela 5 - Tipos de indicadores de implementação de práticas.
Fonte: SEI, 2002.

A primeira etapa no processo de avaliação das ferramentas é o estudo das funcionalidades oferecidas através da utilização da ferramenta e da análise da documentação fornecida pelo fabricante. As evidências que serão coletadas e registradas em uma planilha de avaliação. Esta planilha de avaliação é uma versão adaptada da planilha formal utilizada pelo SCAMPI. Para

visualizar e justificar a adaptação, é mostrada a planilha original, com suas definições e a planilha adaptada, utilizada na avaliação.

Numa avaliação SCAMPI, as evidências obtidas são registradas na planilha de avaliação SCAMPI. Esta planilha é extensa e possui campos específicos para registro e classificação das evidências obtidas, para cada prática de cada área de processo do modelo. Como citado anteriormente, utilizaremos como referência o nível dois de maturidade do modelo.

A planilha de avaliação utiliza as linhas para definir as práticas esperadas pelo modelo e abaixo de cada prática são registradas, pelo avaliador, as evidências obtidas. Após as evidências há uma linha que define se a prática que está sendo avaliada foi implementada. Para ilustrar esta informação, um extrato da planilha pode ser visualizado na figura 16.

Gerenciamento de Requisitos - REQM				
Status		Praticas / Evidências	Origem da Evidência	Documento(s)
	SG1	Os requisitos são gerenciados e as inconsistências com o plano de projeto e produtos de trabalho são identificadas		
	SP1.1-1	Obter um entendimento dos Requisitos		
	1	Evidência 1		
	2	Evidência 2		
	3	Evidência 3		
	4	Evidência 4		
	PF			
	SP1.2-2	Obter Comprometimento dos participantes do projeto com Requisitos		
	1	Evidência 1		
	2	Evidência 2		

Figura 16 - Extrato da planilha de avaliação utilizada pelo SCAMPI.
Fonte: SEI, 2001.

Em função de seu tamanho, a planilha original foi colocada no anexo I. As colunas da planilha definem as informações, para cada evidência coletada. Esta planilha de avaliação possui as seguintes colunas:

- Coluna A - Status – Esta coluna é utilizada pela equipe de avaliação para controle do processo de aprovação da evidência (Revisada, Aprovada etc.).
- Coluna B – Número de identificação único para identificar a evidência.
- Coluna C – Descrição da evidência.

- Coluna D – Identifica a origem da evidência, por exemplo, Líder de Projeto, Desenvolvedor, Gerente Sênior etc.
- Coluna E – identificador único do documento que dá suporte à evidencia descrita.
- Coluna F – Coluna reservada a comentários da equipe de avaliação
- Coluna H – Esta coluna indica (com um "X") se a evidência é atingida a nível organizacional, ou seja, se a evidência faz parte da definição do processo. Usualmente, esta coluna está associada aos objetivos genéricos do modelo.
- Colunas I, J, K, L – Cada coluna indica um dos projetos que está sendo avaliado. Se necessário pode-se adicionar mais colunas. Cada coluna terá o nome do projeto e indica em se a evidência está associada ao projeto.
- Coluna N – Esta coluna indica se a evidência (quando for direta) é forte ou fraca. Marcar com "S"[2] ou "W"[3].
- Coluna O - Esta coluna indica se a evidência (quando for indireta) é forte ou fraca. Marcar com "S" ou "W".
- Coluna P - Esta coluna indica se a evidência (quando for uma afirmação) é forte ou fraca. Marcar com "S" ou "W".
- Coluna R – Colocar um "I" nesta coluna quando é necessário mais informação sobre a implementação da prática. A dúvida deverá ser descrita na coluna B.

Após a última evidência, é colocada a linha PF (*potential findings*), ou achados em potencial. Nesta linha, o avaliador coloca um indicador definindo se a prática foi implementada e em que nível de classificação foi esta implementação. O SCAMPI trará classifica os níveis de implementação em quatro níveis, em acordo com as evidências obtidas e registradas na planilha. Estes níveis de classificação indicam uma pontuação para cada prática avaliada. Os níveis de implementação de uma prática, específica ou genérica, e seu significado são mostrados na figura 17.

[2] A letra "S" vem da palavra *strong* que indica uma evidência forte.
[3] A letra "W" vem da palavra *weak* que indica uma evidência fraca.

Caracterização	Significado
Completamente Implementada *Fully Implemented (FI)*	Artefato(s) direto(s) presente(s) e julgado(s) como adequado(s); Ao menos um artefato indireto e/ou afirmação existente para confirmar a implementação; Nenhum ponto fraco substancial observado.
Largamente Implementada *Largely Implemented (LI)*	Artefato(s) direto(s) presente(s) e julgado(s) como adequado(s); Ao menos um artefato indireto e/ou afirmação existente para confirmar a implementação; Um ou mais pontos fracos substanciais observados.
Parcialmente Implementada *Partially Implemented (PI)*	Artefato(s) direto(s) ausente(s) ou julgado(s) como inadequado(s); Artefatos ou afirmações sugerem que alguns aspectos da prática são implementados; Pontos fracos foram documentados.
Não Implementada *Not Implemented (NI)*	Qualquer outra situação.

Figura 17 - Regras de pontuação utilizadas pelo SCAMPI.
Fonte: SEI, 2001.

Desta forma, na linha PF, para cada projeto avaliado, é colocada a classificação que a prática alcançou. Após a avaliação de todas as práticas específicas e genéricas, é executada a avaliação dos objetivos. A pontuação dos objetivos é feita da seguinte forma:

- O objetivo é considerado SATISFEITO se nenhuma de suas práticas foi classificada abaixo de LI (largamente implementada). Se qualquer prática obtiver uma classificação PI (parcialmente implementada) ou não implementada, o objetivo não é considerado satisfeito.
- O objetivo é considerado INSATISFEITO se houver uma ou mais práticas classificadas como parcialmente implementada ou não implementada.

A próxima etapa é definir a pontuação da área de processo. O resultado da avaliação da área de processo terá como resposta uma pontuação semelhante à utilizada para cada objetivo, seguindo as regras definidas na figura 18.

- Pontuar a satisfação de cada *PA* no escopo da avaliação baseado nas pontuações dos objetivos de cada uma delas. Seguem os tipos de pontuações possíveis:
  - Satisfeita, se e somente se todos os objetivos específicos e genéricos da *PA* são pontuados como satisfeitos;
  - Insatisfeita, se ao menos um dos objetivos da *PA* é pontuado como insatisfeito;
  - Não Aplicável, quando uma *PA* não condiz com o escopo do domínio da aplicação da organização, tal identificação acontece na definição do escopo do modelo na fase de planejamento;
  - Não Pontuada, quando a *PA* está fora do escopo da avaliação, ou quando o conjunto associado de evidências objetivas não atingem os critérios definidos para cobertura mínima de dados necessária para se atribuir qualquer pontuação (caracterização de práticas e pontuação de objetivos).

Figura 18 - Regras de pontuação utilizadas pelo SCAMPI para áreas de processo.
Fonte: SEI, 2006.

Neste contexto, para a avaliação das ferramentas de gerência de requisitos, será feita uma adaptação na planilha uma vez que alguns campos são desnecessários. A avaliação será feita sobre uma ferramenta e não sobre vários projetos de uma organização e serão avaliadas somente as práticas da PA Gerência de Requisitos. A planilha adaptada será instanciada uma vez para cada avaliação de forma que teremos uma planilha para cada ferramenta avaliada. Desta forma, a planilha, adaptada para este fim, é mostrada na tabela 6.

Gerenciamento de Requisitos – REQM

Ferramenta: XXXXXX

Status		Praticas / Anotações	Documento(s)	Comentários		Direta	Indireta	Pontuação
	SG1	**Os requisitos são gerenciados e as inconsistências com o plano de projeto e produtos de trabalho são identificadas**						
	SP1.1-1	**Obter um entendimento dos Requisitos**						
-	1	Evidência 1		-				
-	2	Evidência 2		-				
	3	Evidência 3						
	PF							
	SP1.2-2	**Obter Comprometimento dos participantes do projeto com Requisitos**						
	1	Evidência 1						
	2	Evidência 2		-				
	3	Evidência 3		-				
	PF							

	SP1.3-1	**Gerenciar Mudanças de Requisitos a medida que elas ocorram durante o projeto**						
	1	Evidência 1						
	2	Evidência 2						
	3	Evidência 3						
	PF							
	SP1.4-2	**Manter Rastreabilidade Bidirecional entre os Requisitos e o plano de projeto e demais artefatos do projeto**						
	1	Evidência 1						
	2	Evidência 2						
	3	Evidência 3						
	PF							
	SP1.5-1	**Identificar Inconsistências entre o plano do projeto e demais artefatos e os requisitos**						
	1	Evidência 1						
	2	Evidência 2						
	3	Evidência 3						
	PF							
	GG2	**Institucionalizar um processo gerenciado**						
	GP2.1	**Estabelecer e manter uma política organizacional para o planejamento e execução do processo de gerenciamento de requisitos.**						
	1	Evidências						
	PF							
	GP2.2	**Estabelecer e manter o plano para a execução do processo de gerenciamento de requisitos.**						
	1	Evidências						
	PF							
	GP2.3	**Fornecer os recursos adequados para a execução do processo de gerenciamento de requisitos, desenvolvimento de produtos de trabalho e fornecimento dos serviços do processo.**						
	1	Evidências						
	PF							
	GP2.4	**Atribuir responsabilidades e autoridade para a execução do processo, desenvolvimento dos produtos de trabalho e fornecimento dos serviços do processo de gerenciamento de requisitos.**						
	1	Evidências						
	PF							
	GP2.5	**Treinar as pessoas para executar e suportar o processo de gerenciamento de requisitos, conforme necessário.**						
	1	Evidências						
	PF							
	GP2.6	**Colocar os produtos de trabalho definidos do processo de gerenciamento de requisitos sob os níveis apropriados de gerenciamento de configurações.**						
	1	Evidências						
	PF							
	GP2.7	**Identificar e envolver os stakeholders relevantes do processo de gerenciamento de requisitos, conforme planejado.**						
	1	Evidências						

	PF							
	GP2.8	Monitorar e controlar o processo de gerenciamento de requisitos contra o plano de execução do processo e tomar a ação corretiva apropriada.						
	1	Evidências						
	PF							
	GP2.9	Avaliar objetivamente a aderência do processo de gerenciamento de requisitos contra sua descrição de processo, padrões e procedimentos e tratar as não conformidades.						
	1	Evidências						
	PF							
	GP2.10	Revisar as atividades, status e resultados do processo de gerenciamento de requisitos com o nível mais alto nível de gerência e resolver questões.						
	1	Evidências						
	PF							

Tabela 6 - Planilha de avaliação de ferramentas de gerência de Requisitos.

Esta planilha tem, em suas linhas, a distribuição das práticas propostas pelo CMMI e, abaixo de cada prática, linhas para descrição das evidências encontradas nas ferramentas. Da mesma forma, após as evidências de cada prática há a linha PF, onde será colocada a pontuação obtida pela ferramenta. As colunas utilizadas por esta planilha têm a mesma função da planilha original, porém diversas colunas foram removidas por terem sido consideradas desnecessárias no contexto desta avaliação. A fim de justificar a adaptação, descreve-se, na tabela 7 as colunas que foram removidas e sua justificativa.

Coluna removida	Justificativa
Coluna D – Identifica a origem da evidência, por exemplo, Líder de Projeto, Desenvolvedor, Gerente Sênior etc.	Esta coluna foi removida, pois todas as evidências foram coletadas pelo autor deste trabalho.
Coluna H – Esta coluna indica (com um "X") se a evidência é atingida a nível organizacional, ou seja, se a evidência faz parte da definição do processo.	Esta coluna foi removida pois será avaliada uma ferramenta, de modo que está informação está fora do escopo.
Colunas I, J, K, L – Cada coluna indica um dos projetos que está sendo avaliado. Se necessário pode-se adicionar mais colunas. Cada coluna terá o nome do projeto e indica em se a evidência está associada ao projeto.	Estas colunas foram removidas pois será avaliada uma ferramenta, de forma que não há projetos a serem avaliados.
Coluna P - Esta coluna indica se a evidência (quando for uma afirmação) é forte ou fraca. Marcar com "S" (forte) ou "W" (fraca).	Esta coluna foi removida uma vez que não será feita coleta de evidências com base em afirmações. As evidências serão coletadas através da análise da ferramenta e da documentação fornecida pelo fabricante.
Coluna R – Colocar um "I" nesta coluna quando é necessário mais	Em função de ser uma avaliação individual sobre as ferramentas de gerência de requisitos e esta avaliação ser feita somente pelo autor deste

informação sobre a implementação da prática. A dúvida deverá ser descrita na coluna B.	trabalho, optou-se por remover esta coluna de dúvidas.

Tabela 7 - Colunas removidas da planilha de avaliação.

A avaliação das ferramentas é feita pelas atividades de estudo da ferramenta e sua documentação a procura de evidências que indiquem se a ferramenta atende às práticas indicadas pelo modelo. Estas evidências poderão ser artefatos diretos ou indiretos. Após o estudo da ferramenta, as evidências encontradas serão registradas na planilha de avaliação e pontuadas segundo as mesmas regras utilizadas pelo SCAMPI, com algumas adaptações. As evidências encontradas nas ferramentas serão definidas como:

- DIRETAS: se e somente se a ferramenta automatiza completamente o artefato que evidencia a implementação da prática. Como exemplo, pode-se citar o recurso de rastreabilidade proporcionado pela ferramenta Rational RequisitePro (que será avaliada nas próximas sessões). Da mesma forma, se a evidência é direta, ela será considerada FI (*Fully Implemented*). A figura 19 mostra uma tela exemplo onde é criada a associação de rastreabilidade e uma tela exemplo onde se pode ver a matriz de rastreabilidade oferecida por uma das ferramentas que serão avaliadas.

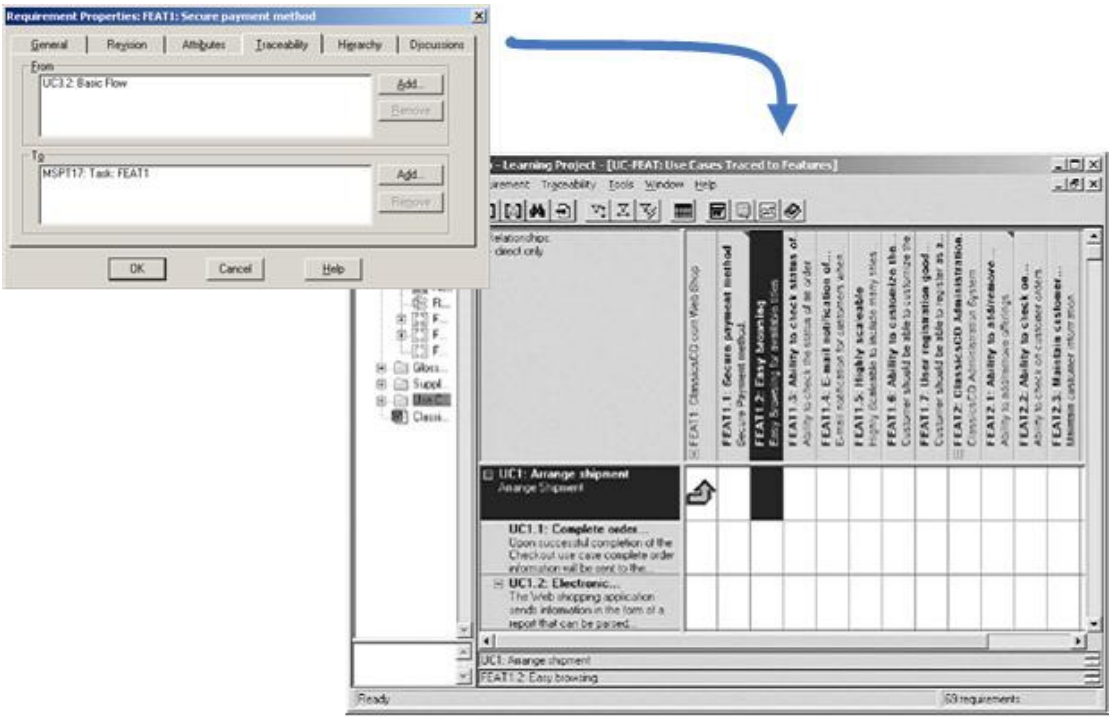


Figura 19 - Tela de associação de rastreabilidade e matriz de rastreabilidade.

- INDIRETAS: a evidência será indireta se a ferramenta oferece um *template* para preenchimento das informações ou se a ferramenta necessita de customizações para que a

prática seja atendida. Da mesma forma, há a necessidade de avaliar os *templates* oferecidos ou as customizações a serem feitas para definir se a prática será LI (*Largely Implemented*) ou PI (*Partially Implemented)*. Ou seja, se os *templates* oferecidos atendem completamente as práticas, teremos uma evidência indireta com uma pontuação LI, se os *templates* atendem parcialmente as práticas, teremos uma evidência indireta com uma pontuação PI. Como exemplo pode-se citar os *templates* oferecidos pela ferramenta Rational RequisitePro, que são modelos do RUP (*Rational Unified Process*). Outro exemplo são as customizações que pode ser implementada na ferramenta Borland CaliberRM que permitem criar artefatos para apoiar diversas práticas. A figura 20 mostra um exemplo de um checklist customizado na ferramenta CaliberRM e, na figura 21, um exemplo de *template* oferecido pela ferramenta Rational RequisitePro para definição de casos de uso.

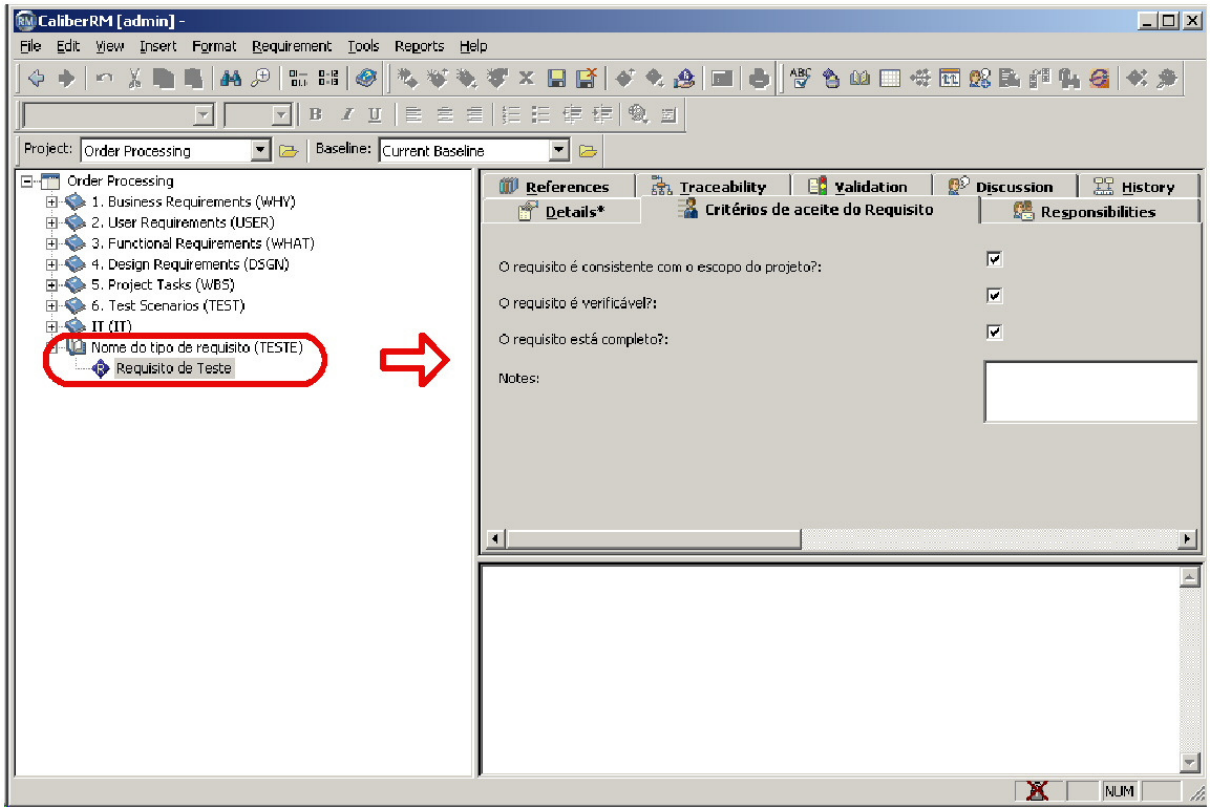
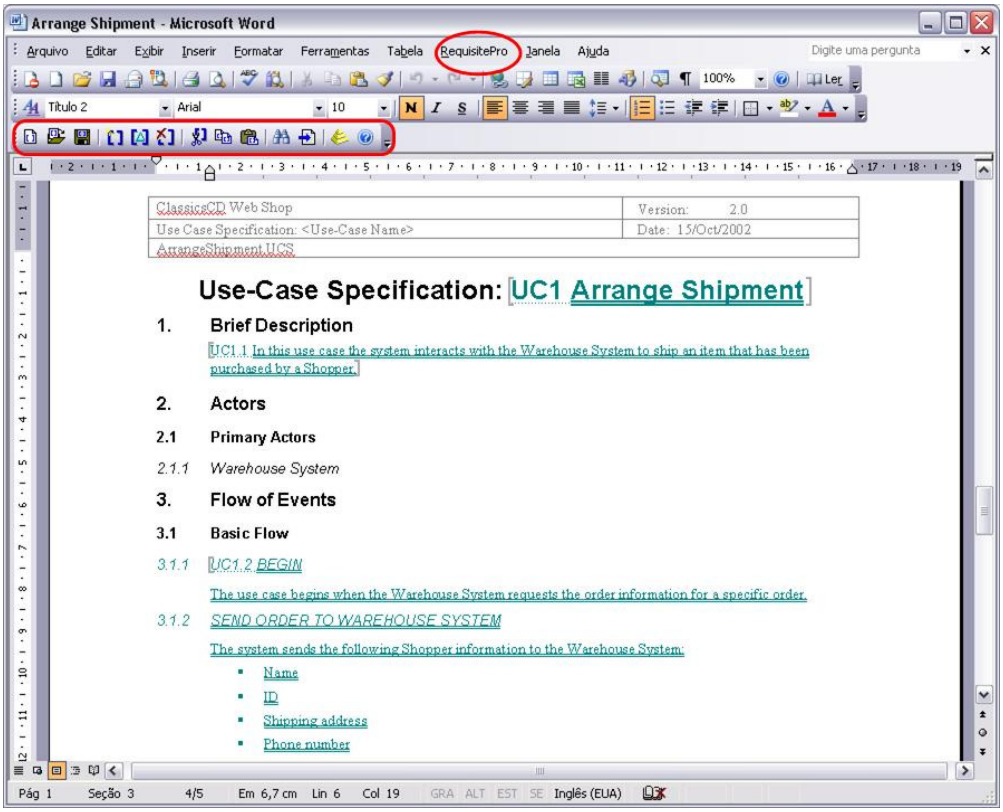


Figura 20 - Checklist customizado na ferramenta Borland CaliberRM.


Arrange Shipment - Microsoft Word
Arquivo Editar Exibir Inserir Formatar Ferramentas Tabela RequisitePro Janela Ajuda
ClassicsCD Web Shop
Version: 2.0
Use Case Specification: <Use-Case Name>
Date: 15/Oct/2002
ArrangeShipment.UCS
Use-Case Specification: UC1 Arrange Shipment
1. Brief Description
UC1.1 In this use case the system interacts with the Warehouse System to ship an item that has been purchased by a Shopper.
2. Actors
2.1 Primary Actors
2.1.1 Warehouse System
3. Flow of Events
3.1 Basic Flow
3.1.1 UC1.2 BEGIN
The use case begins when the Warehouse System requests the order information for a specific order.
3.1.2 SEND ORDER TO WAREHOUSE SYSTEM
The system sends the following Shopper information to the Warehouse System:
Name
ID
Shipping address
Phone number
Figura 21 - *Template* para especificação de casos de uso no Rational RequisitePro.

Caso a ferramenta não ofereça nenhum artefato direto nem *templates* nem a opção de customizações, a prática será considerada como NI (*Not Implemented*). Além disso, dentro do contexto acadêmico deste trabalho, será utilizada, na planilha adaptada, a coluna comentários para justificar a pontuação alcançada por cada evidência obtida.

Nas próximas sessões serão apresentadas as avaliações feitas em duas ferramentas de gerência de requisitos em evidência no mercado que são o CaliberRM, da Borland, e o RequisitePro, da Rational. Além destes, será aplicada a avaliação sobre a plataforma PLACES, de forma a definir quais práticas são atendidas e quais práticas não são atendidas. Sobre as práticas não atendidas pelo PLACES será criada uma lista de melhorias a serem implementadas na segunda etapa deste trabalho para buscar o alinhamento da ferramenta ao modelo CMMI.

5.3 AVALIAÇÃO DA FERRAMENTA CALIBERRM

5.3.1 Descrição da Ferramenta

O CaliberRM[4] é uma ferramenta para gerência de requisitos desenvolvida pela Borland, cujo objetivo é elicitar, documentar, gerenciar e acompanhar os requisitos durante todo o ciclo de vida do projeto, permitindo que as equipes de desenvolvimento de software avaliem o impacto das constantes mudanças de requisitos, fornecendo prontas-respostas, sem que isso coloque o sucesso do projeto em risco.

A ferramenta se propõe a gerenciar a mudança nos requisitos e manter um sistema de comunicação entre os stakeholders do projeto que são afetados por estas mudanças. Os requisitos são armazenados em um banco de dados para prover um conjunto atualizado disponível para os integrantes do projeto. O Borland CaliberRM é composto pelos componentes listados na tabela 8.

Componente	Descrição
CaliberRM	Sistema de gerência de requisitos
Document Factory	Ferramenta para customização de documentos para a criação de *templates* destinados aos usuários finais.
Framework Administrator	Ferramenta para controle de funções administrativas do projeto. Nesta ferramenta é feito o cadastramento dos usuários e grupos, bom como algumas atribuições de responsabilidade destes usuários e grupos no projeto.
Requeriment Viewer	Ferramenta cliente utilizada para visualização dos requisitos

Tabela 8 - Componentes da ferramenta CaliberRM.
Fonte: Borland, 2006.

O CaliberRM trabalha com o conceito de projetos. Ao projeto são associados usuários, artefatos (requisitos, casos de uso etc.) e documentos externos para gerenciamento centralizado.

O CaliberRM, definido pelo fabricante como ferramenta para gerenciamento de requisitos de um projeto, faz parte da família de produtos Caliber, que possui, ainda o produto Caliber DefineIT, para definição de requisitos e o Caliber Analyst, para integração do CaliberRM e Caliber DefineIT, acrescido de recursos para trabalho colaborativo com equipes distribuídas. Na figura 22 é mostrada a interface do CaliberRM.

[4] Borland CaliberRM, é um software para gerência de requisitos desenvolvido pela Borland. Atualmente em sua versão 2005-SP2, ele pode ser baixado, em versão trial (roda por 30 dias), de www.borland.com.

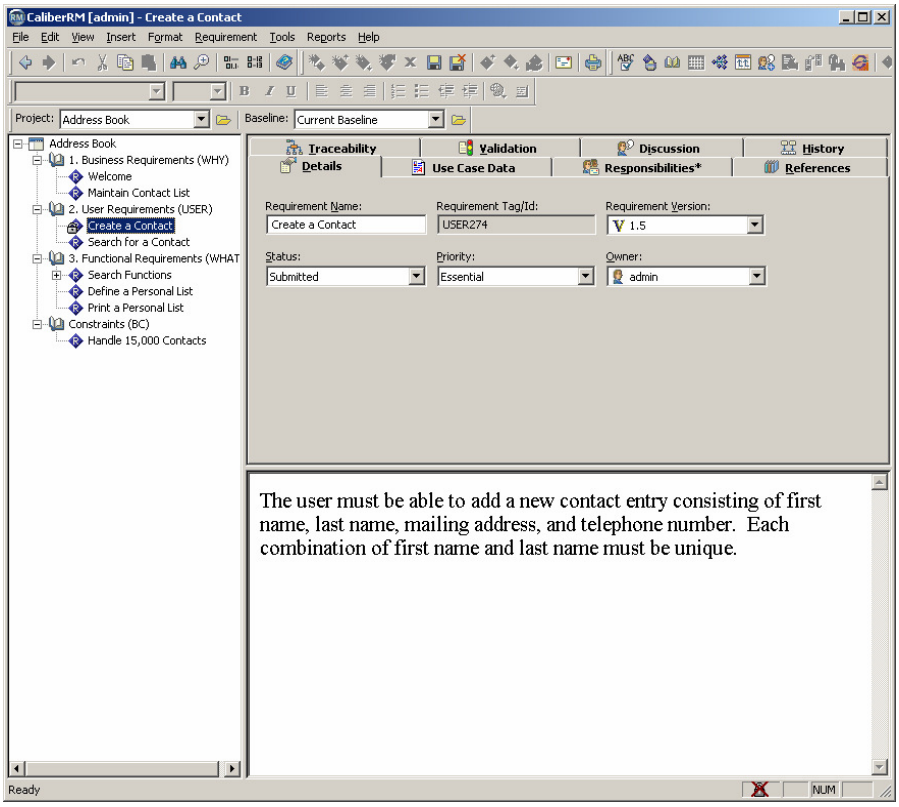


Figura 22 - Interface da ferramenta Borland CaliberRM.

Dentre as principais características da ferramenta, destacam-se:

- **Repositório central**

O Borland CaliberRM fornece um repositório central e seguro para os requisitos do projeto. Este repositório pode ser no banco de dados próprio da aplicação ou utilizar bancos de dados Oracle ou Microsoft SQLServer.

- **Adaptável aos processos da organização**

Esta adaptabilidade é caracterizada pela possibilidade de criação de tipos de requisitos com atributos customizados. Desta forma pode-se criar atributos do tipo "*use case*" (casos de uso) com atributos customizados que suportem as informações dos casos de uso utilizadas pela empresa. O software traz alguns tipos considerados padrõnizados, porém o próprio fabricante indica que deve ser feita uma customização para a adaptação total ao processo da empresa. Na figura 23 é mostrada as etapas de criação de um tipo de requisito e, na figura 24, como este tipo de requisito é usado no projeto.

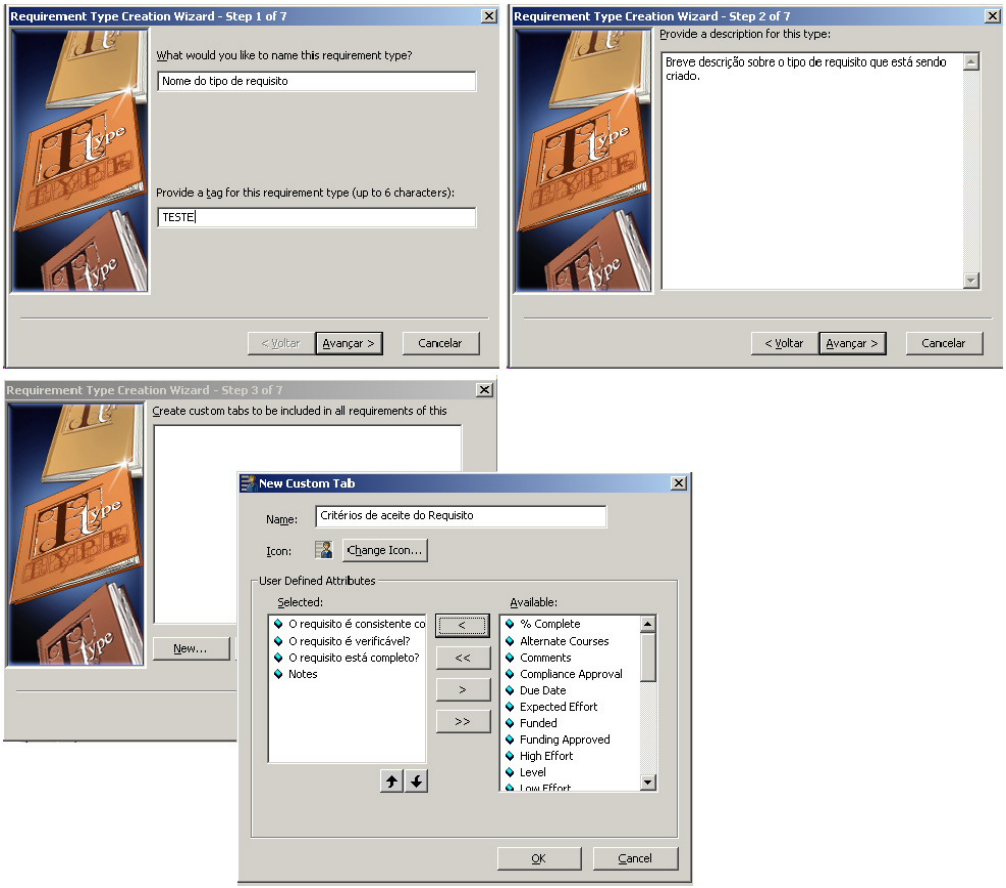


Figura 23 - Criação de um tipo de requisito CaliberRM.

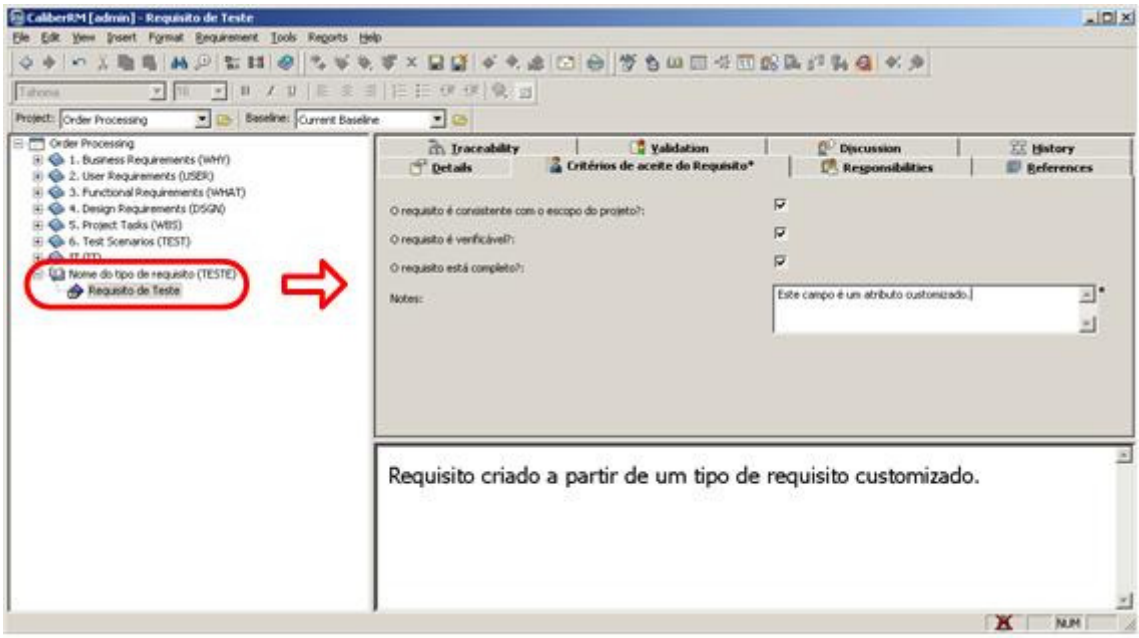


Figura 24 - O Requisito criado sendo utilizado no projeto.

Também é possivel a criação de usuários e grupos de usuários, com permissões diferenciadas para cada tipo de requisito existente. Desta forma é possivel definir quem tem autorização

para executar as tarefas de revisão e aprovação de requisitos. Esta configuração pode ser feita por projeto. Nas figuras 25 e 26 são mostradas as telas de criação de usuário e grupo.

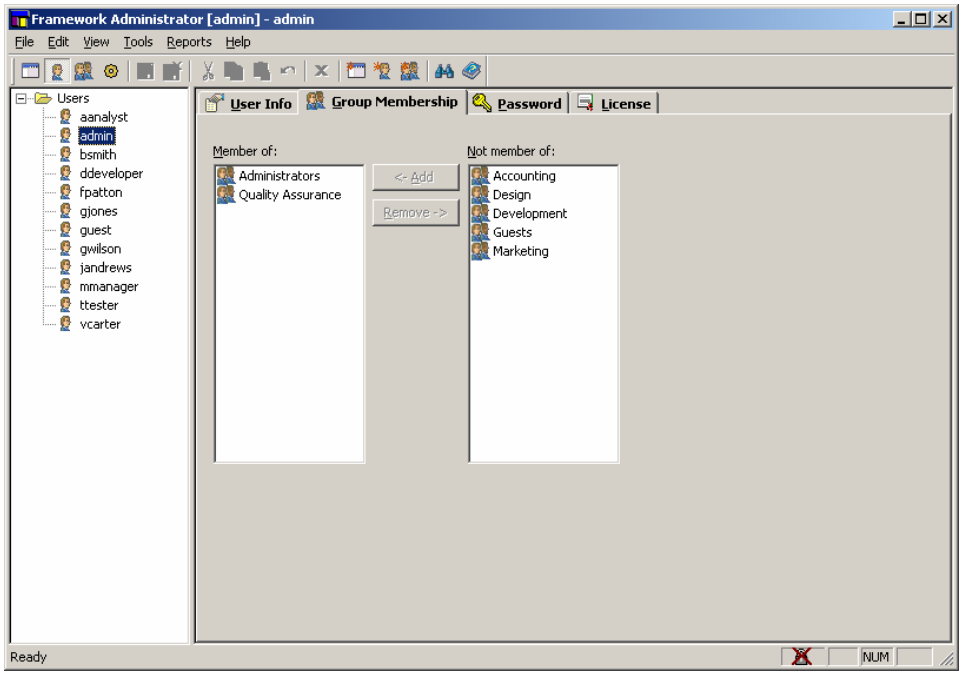


Figura 25 - Criação de usuário no CaliberRM.

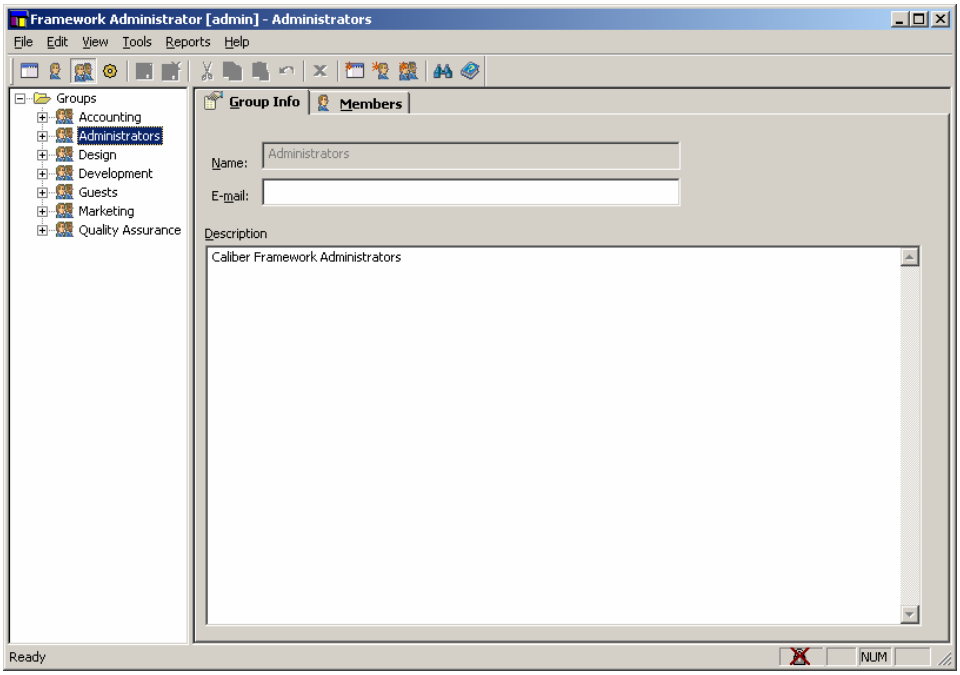


Figura 26 – Criação de grupo de usuários no CaliberRM.

- **Rastreamento de requisitos durante todo ciclo de vida do projeto**

A ferramenta possui arquitetura aberta no sentido de permir que os requisitos de software sejam ligados a uma variedade de artefatos em todo o ciclo de vida. Esta ligação pode ser

direta a outros requisitos ou utilizando-se de *drivers* de acesso às artefatos em outras aplicações (por exemplo, tarefas de um projeto no MSProject). Há ainda a possibilidade de criar rastreabilidade para documentos externos. A figura 27 mostra a tela onde é definida a rastreabilidade do tipo de requisito. No quadro *Traces to* indica que artefatos são dependentes do requisito e no quadro *Traces from* indica quais artefatos o requisito é dependente. Para indicação que houve alguma alteração na associação de rastreabilidade há o campo "status" que é marcado como "suspect" indicando alguma alteração no artefato. Observar que a caixa "coment", que apareca à direita na tela, é instanciada toda a ves que alguma alteração é feita em algum requisito. O texto digitado na caixa, fica registrado no histórico do requisito com o objetivo de manter um histórico das alterações que ocorreram no mesmo.

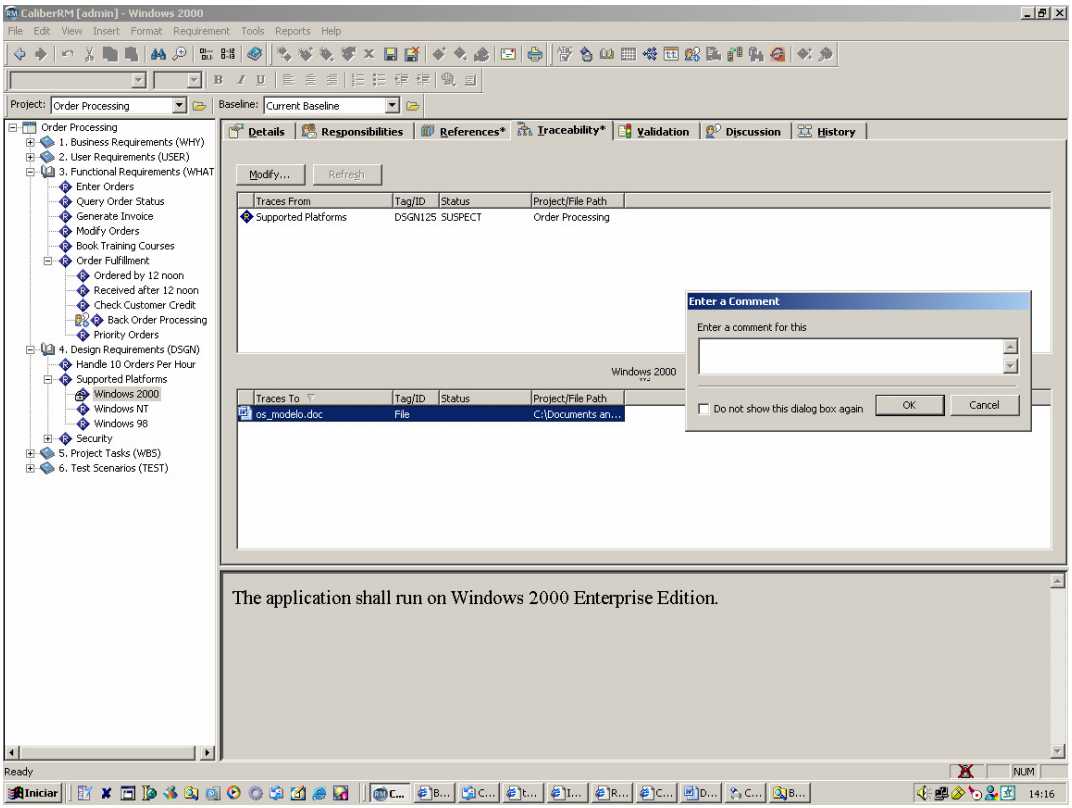


Figura 27 - Definição das associações de rastreabilidade do requisito no CaliberRM.

A figura 28 mostra a ligação de um requisito a uma tarefa em um projeto do MSProject. Observar que é criada uma *TAB* especial contendo os valores da tarefa do MSProject.

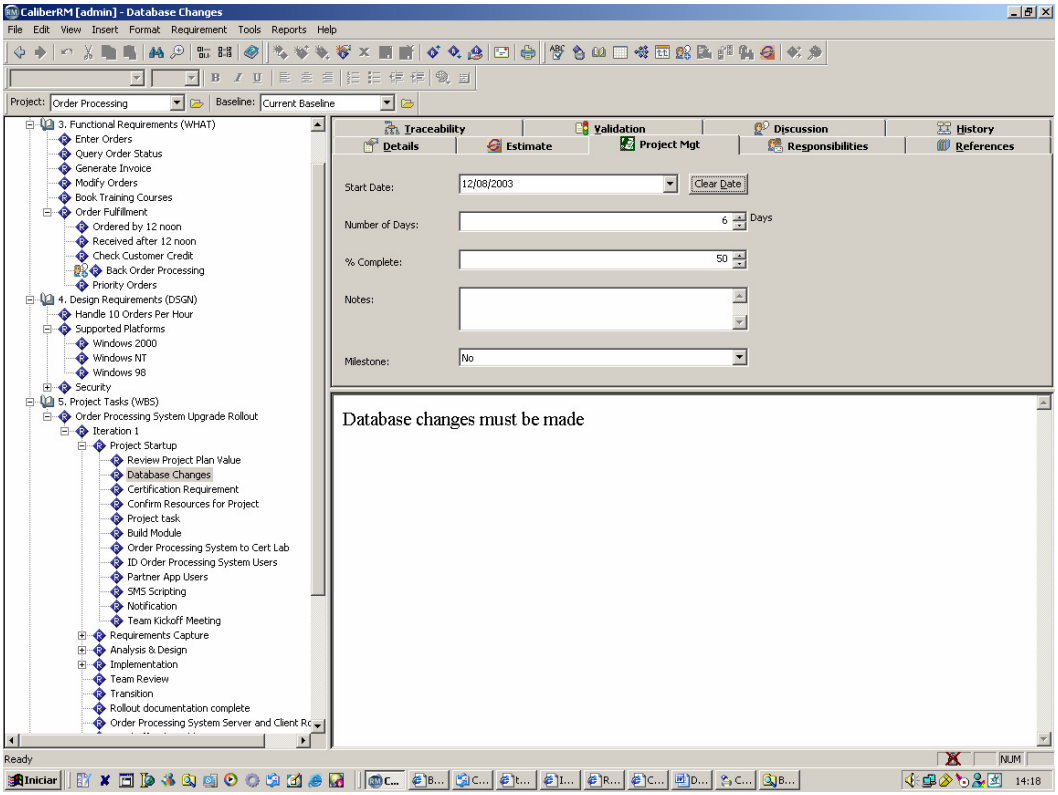


Figura 28 - Integração dos requisitos com tarefa do projeto, controlado no MSProject.

- **Análise de impacto durante todo o ciclo de vida da aplicação**

A ferramenta possui múltiplos métodos de visualização da rastreabilidade, ajudando os usuários a compreender o escopo de análise necessário para que se tenha uma previsão dos efeitos de uma determinada mudança de requisitos. A ferramenta apresenta dois artefatos para visualização de rastreabilidade que são a matriz de rastreabilidade e o diagrama de rastreabilidade. Tanto a matriz de rastreabilidade é gerada a partir de um conjunto pré-definido de tipos de requisitos que é definido quando se solicita o recurso. Da mesma forma pode-se filtrar estes tipos de artefatos por seus status, stakeholders responsáveis ou qualquer outro atributo, inclusive atributos customizados.

A ferramenta permite criar matrizes de rastreabilidade entre quaisquer tipos de requisitos ou artefatos definidos para o projeto. A figura 29 mostra uma matriz de rastreabilidade mostrando a relação entre casos de uso e requisitos funcionais.

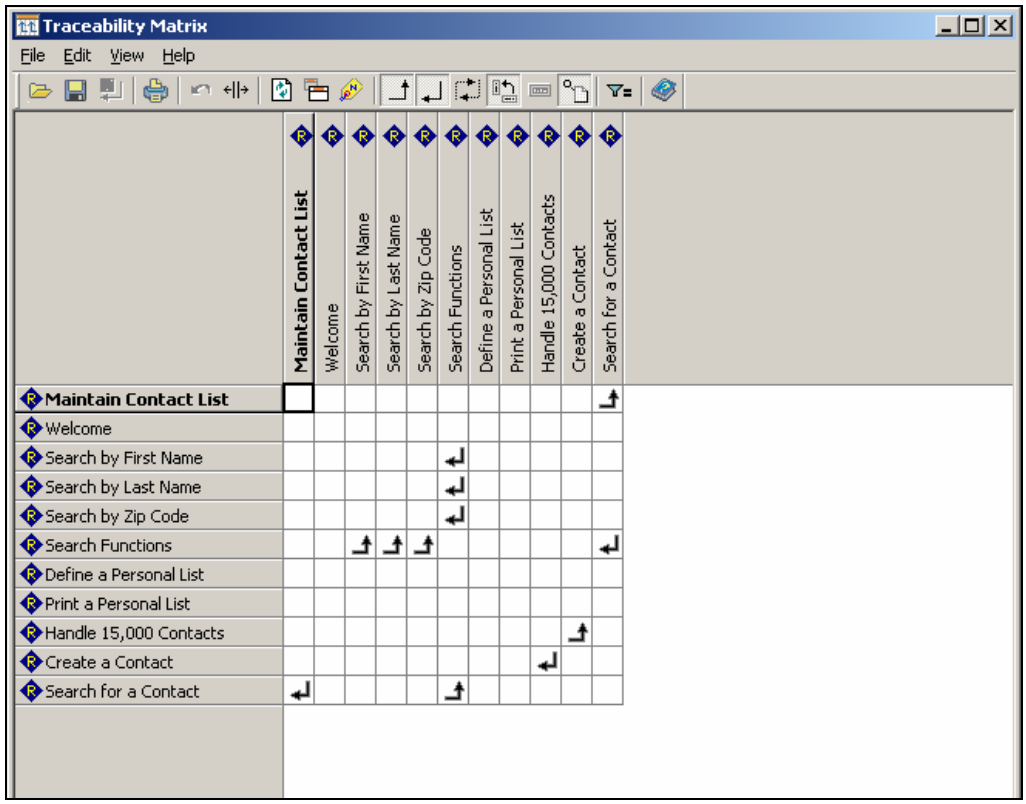


Figura 29 - Matriz de rastreabilidade do CaliberRM.

Os diagramas de rastreabilidade mostram as ligações a partir de um artefato definido. Esta forma de visualização permite observar todos os artefatos que dependem do requisito e todos os artefatos que o requisito é dependente. A figura 30 mostra um diagrama de rastreabilidade mostrando as relações de dependência de um requisito.

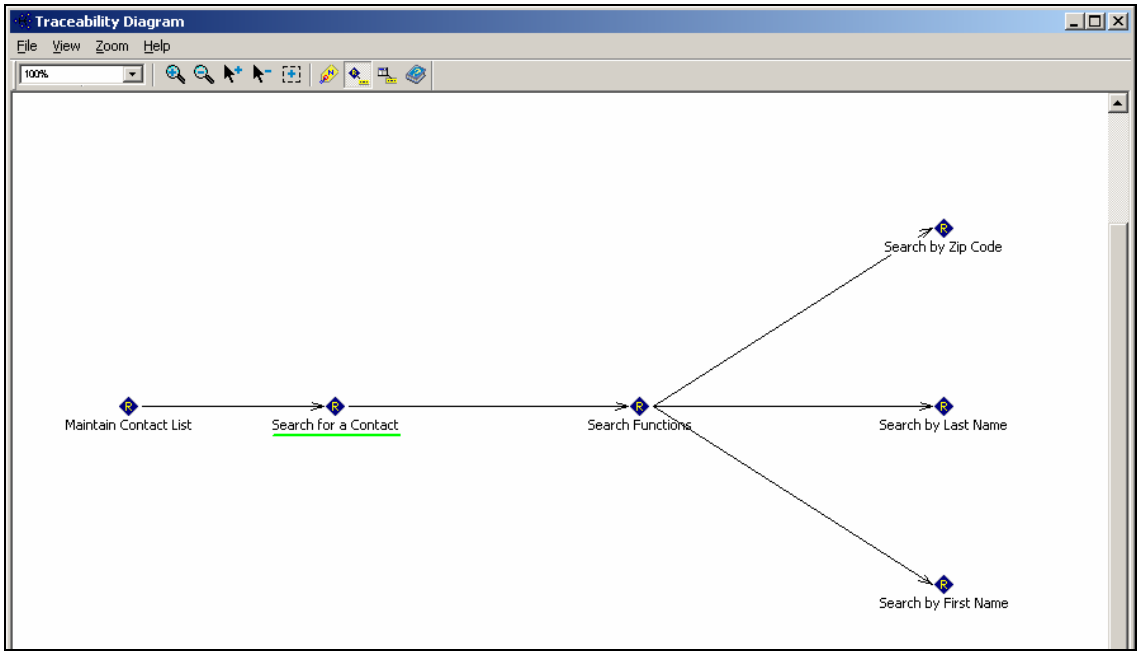


Figura 30 - Diagrama de rastreabilidade do CaliberRM.

- **Glossários *on-line* para padronização e definição da terminologia**

Os glossários são úteis na definição de termos da indústria, referências de projeto, linguagem corporativa etc. – para ajudar a assegurar que todos os membros da equipe tenham uma mesma percepção dos requisitos. A ferramenta permite a criação de vários glossários e estes glossários são atribuídos aos projetos por relevância. A figura 31 mostra a tela de criação de glossários.

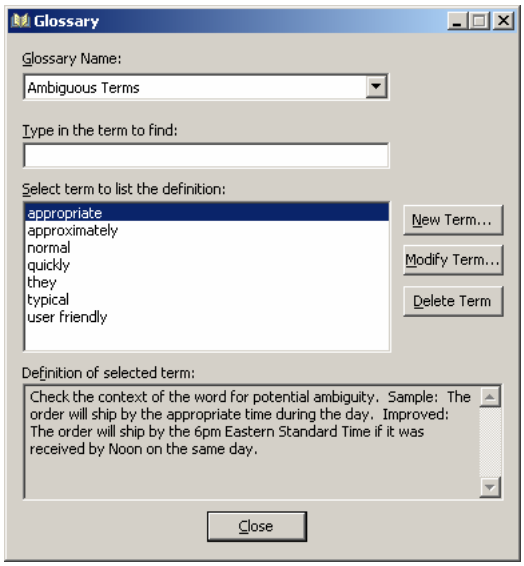

Glossary
Glossary Name:
Ambiguous Terms
Type in the term to find:
Select term to list the definition:
appropriate
approximately
normal
quickly
they
typical
user friendly
New Term...
Modify Term...
Delete Term
Definition of selected term:
Check the context of the word for potential ambiguity. Sample: The order will ship by the appropriate time during the day. Improved: The order will ship by the 6pm Eastern Standard Time if it was received by Noon on the same day.
Close

Figura 31 - Criação de glossários no CaliberRM.

A figura 32 mostra a tela de inclusão de um termo no glossário.

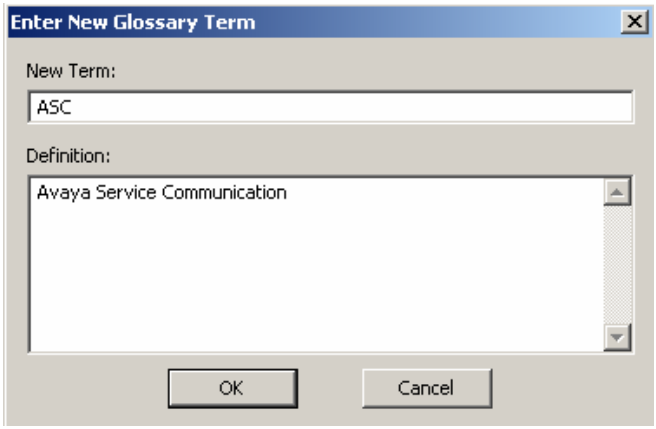


Figura 32 - Criação de termos no glossário no CaliberRM.

O CaliberRM notifica, via e-mail, os usuários designados sobre as alterações que são feitas em um requisito. Da mesma forma, possui um recurso chamado “discussão” onde os usuários podem deixar dúvidas sobre o sobre o requisito para serem respondidas. Uma tela com um exemplo pode ser vista na figura 33.

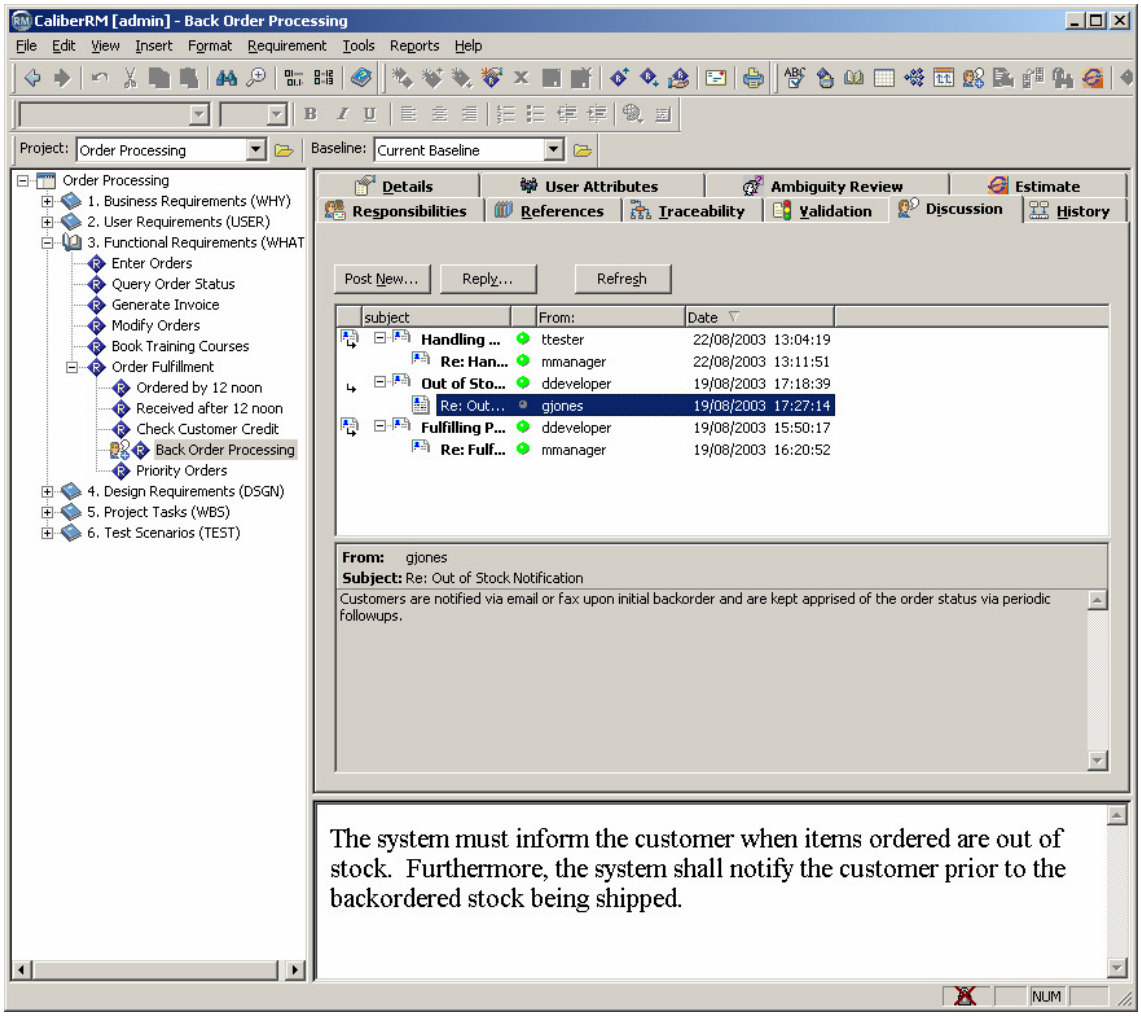


Figura 33 - Tela de discussões sobre o entendimento dos requisitos no CaliberRM.

Outro recurso da ferramenta é a facilidade de atribuição direta dos requisitos aos usuários, definindo assim os responsáveis pelos mesmos. Na figura 34, é mostrada a interface de atribuição da responsabilidade de um requisito para um usuário cadastrado no sistema. Pode-se atribuir responsabilidade de um requisito para um usuário ou para um grupo de usuários. Desta forma, quando um grupo é definido como responsável por um requisito todos os usuário deste grupo terão as mesmas permissões e também todos receberão notificações quando houver alguma alteração sobre o requisito.

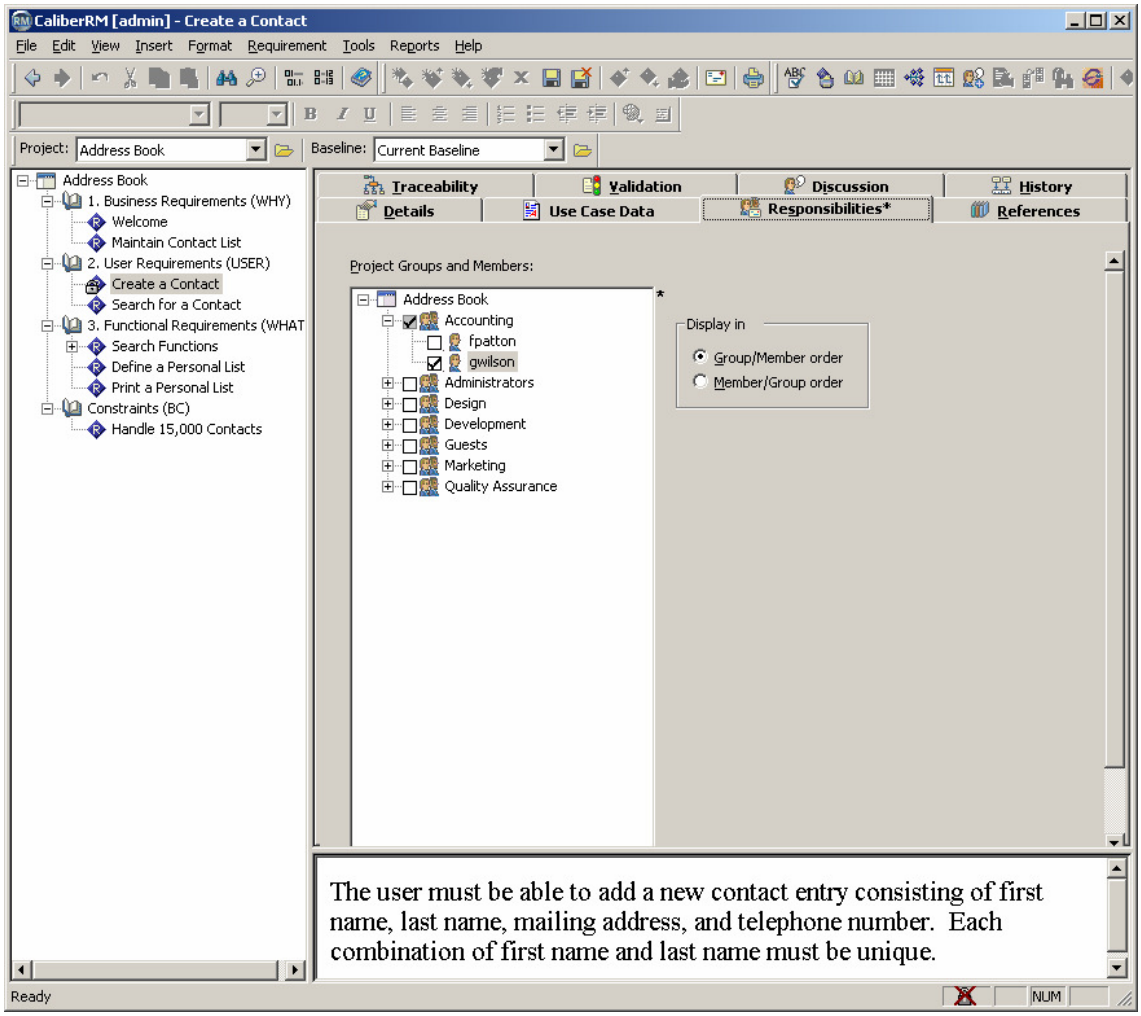


Figura 34 - Tela de atribuição de responsabilidades dos requisitos no CaliberRM.

A ferramenta oferece recursos para criação e manutenção de *baselines* do projeto. Estas *baselines* incluem todos as artefatos do projeto e podem ser configuradas para ter a necessidade de aprovação por usuários com permissão para este recurso. Através de um *framework* interno é possíveis definir os aprovadores da *baseline.* Outro recurso interessante é que é possível bloquear a *baseline* durante o processo de revisão de forma a manter íntegras as informações que a compõe. Caso a *baseline* não seja aprovada, é possível desbloqueá-la para alterações ou simplesmente excluir suas referencias nos artefatos do projeto. A figura 35 mostra a tela de criação e manutenção de *baselines*.

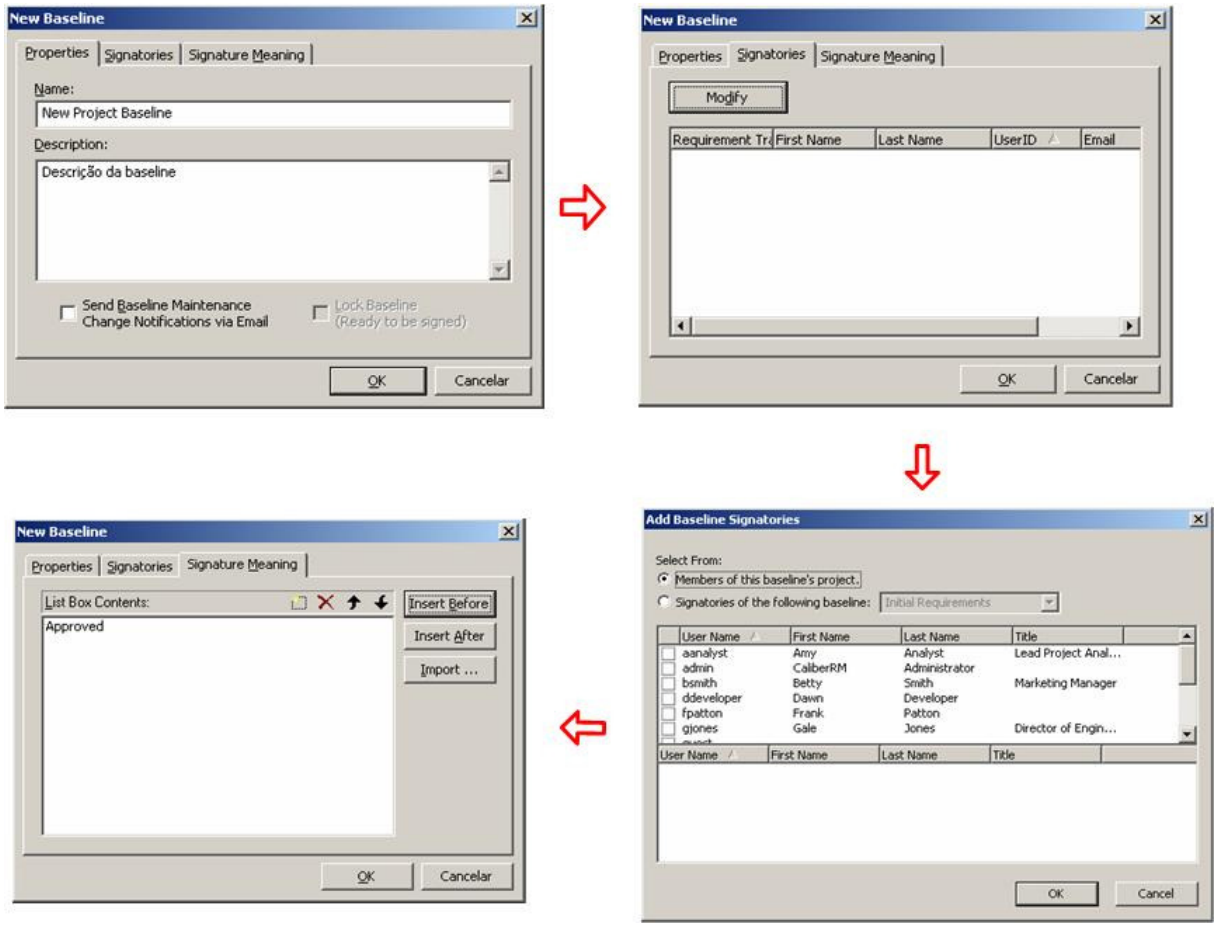


Figura 35 - Criação de *baselines* no CaliberRM.

Para a documentação de requisitos, a ferramenta gera documentos de requisitos e demais documentos customizados a partir de *templates* criados pelo componente "*Document Factory*". O CaliberRM não oferece nenhum template pré-definido. Segundo o fabricante, uma das etapas de implantação da ferramenta passa pela criação dos *templates* para geração de documentos a partir do "*Document Factory*". Para dar suporte à customização de *templates*, a ferramenta oferece uma série de comandos a serem utilizados na criação do template de maneira que qualquer informação cadastrada no projeto pode ser utilizada como fonte de dados para a geração automática do documento. Este recurso de integração do CaliberRM com o Word é disponibilizado pelo CaliberRM SDK (*Software Development Kit*), que é distribuído com a ferramenta. Este SDK disponibiliza, também, interfaces (através de bibliotecas de vínculo dinâmico) para que seja possível desenvolver ferramentas externas para acessar dados dos projetos.

A ferramenta "*Document Factory*" gera documentos no formado do editor Word, da Microsoft, porém é possível abrir os documentos a partir de qualquer editor compatível com este tipo de documento. A fonte dos dados para os documentos é gerada a partir de visões dos

requisitos onde é possível aplicar filtros por qualquer atributo do requisito. Desta forma é possível gerar qualquer padrão de documentação impressa a partir dos dados armazenados no banco de dados da aplicação. A figura 36 mostra um *template* de documento, em acordo com Wiegers (2001) obtido no *site* do fabricante onde se pode observar o estilo de linguagem que é utilizado para obter os dados para o documento.

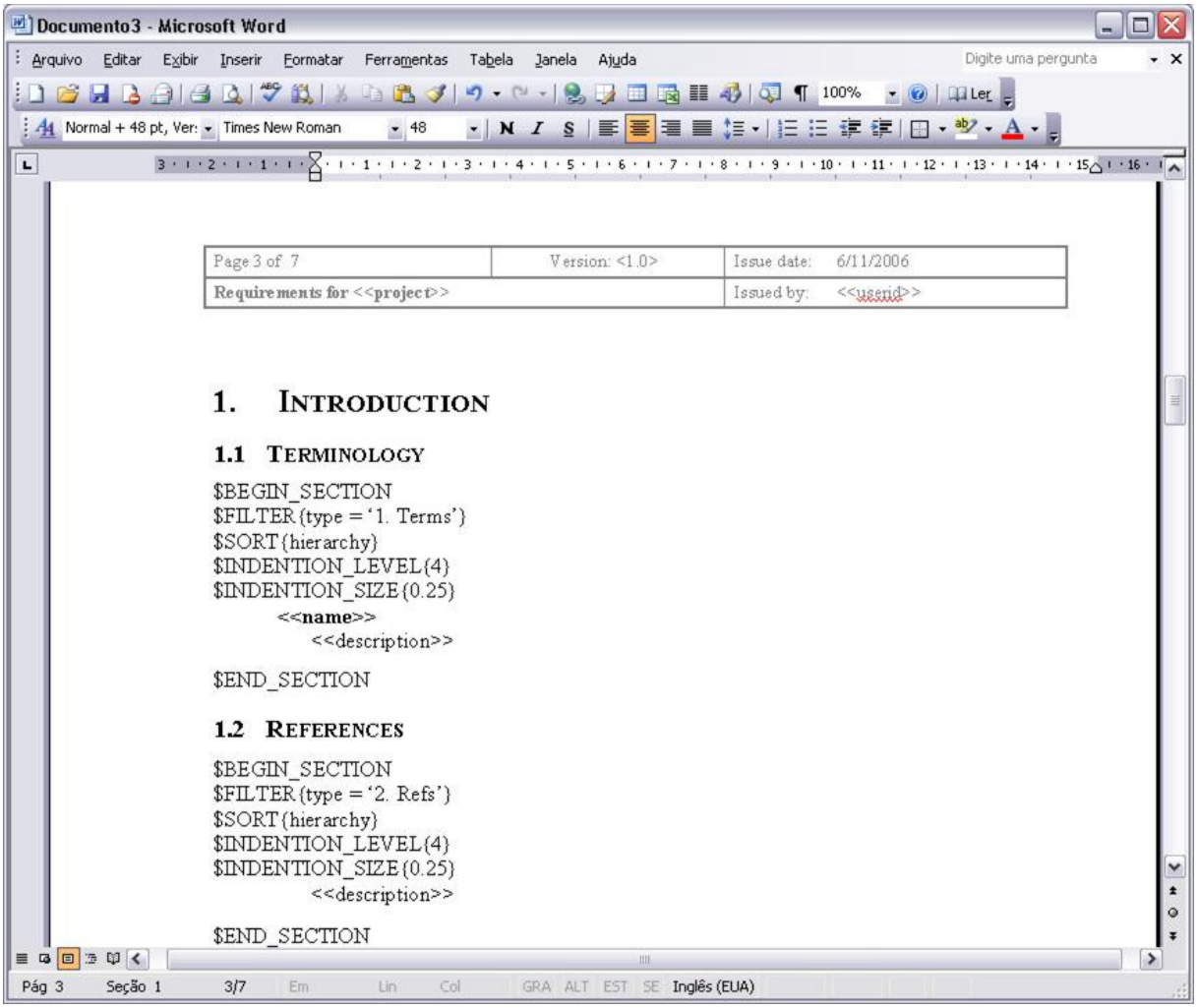


Figura 36 - Template de documento do CaliberRM.

Como componente opcional, o fornecedor disponibiliza uma interface *web* para visualização dos requisitos do projeto. Este recurso permite o trabalho de equipes distribuídas num mesmo projeto.

O CaliberRM possui interface intuitiva, é fácil de utilizar e possui muitas funcionalidades de suporte à decisão, previsibilidade e análise de impacto, além de utilizar recursos de correio eletrônico permitindo que todos os envolvidos no projeto colaborem entre si durante o ciclo

de vida do projeto. O fabricante da ferramenta indica que a ferramenta é aderente ao nível dois do CMMI desde que seja utilizada em conjunto com outras ferramentas. A orientação do fabricante é o uso do CaliberRM em conjunto com as outras ferramentas para que se tenha um alinhamento completo com o CMMI. De qualquer forma, é interessante ressaltar que o fabricante deixa claro que é possível atingir o nível dois com o uso da ferramenta CaliberRM associada a ferramentas de outros fabricantes, ficando a cargo da empresa customizar o uso e a integração da aplicação de gerência de requisitos com os demais aplicativos utilizados.

5.3.2 Aplicação da avaliação

A avaliação foi feita sobre o Borland CaliberRM 2005 sp2, baixado de www.borland.com em abril de 2005. O processo de instalação da ferramenta não será colocado neste trabalho em função de ser um processo intuitivo e bem documentado pelo fabricante. Para a execução da avaliação foi utilizado o projeto exemplo que vem pré-instalado no banco de dados da aplicação. Segundo o fabricante este projeto utiliza os recursos básicos da ferramenta.

Como referência para aprendizado da ferramenta utilizou-se a documentação instalada junto com o produto. Esta documentação mostrou-se completa, suprindo todas as dúvidas e possibilidades de implementações necessárias para a execução desta avaliação.

A tabela 9 mostra os resultados da avaliação e registra as evidências que foram encontradas para suprir as práticas indicadas na PA Gerência de Requisitos do CMMI. Apesar do CaliberRM não fornecer *templates* para geração de documentos, a coluna "Documentos" foi deixada na planilha para que não se tenha uma planilha específica para a avaliação de cada ferramenta.

PA Gerência de Requisitos - REQM

Ferramenta: Borland Caliber RM

Status		Praticas / Anotações	Documento(s)	Comentários		Direta	Indireta	Pontuação
	SG1	**Os requisitos são gerenciados e as inconsistências com o plano de projeto e produtos de trabalho são identificadas**						
	SP1.1-1	**Obter um entendimento dos Requisitos**						
	1	**A ferramenta oferece componente para criação de *templates*** para criação de critérios de aceite de fornecedores de requisitos					**X**	**LI**

	2	**A customização de atributos** de requisitos permite a criação de *check-lists* para aceite dos requisitos.					X	LI
	3	Através do **framework configurável**, pode-se criar grupos de avaliadores para dar o aceite dos requisitos.				X		FI
	4	A ferramenta oferece um **recurso chamado "discussão"** que cria um ambiente colaborativo para que todos os envolvidos obtenham um entendimento comum quanto à definição dos requisitos.					X	LI
	PF							LI
	SP1.2-2	**Obter Comprometimento dos participantes do projeto com Requisitos**						
	1	**Os requisitos definidos são atribuídos** aos stakeholders em campo definido para este fim. **O usuário é solicitado a efetuar seu aceite.**				X		FI
	PF							FI
	SP1.3-1	**Gerenciar Mudanças de Requisitos a medida que elas ocorram durante o projeto**						
	1	A ferramenta **notifica os usuários por e-mail** designados quando há mudanças nos requisitos.				X		FI
	2	Controle da situação da solicitação da mudança através de **status** que indica se o mesmo foi alterado, aprovado ou rejeitado.				X		FI
	3	Matriz de rastreabilidade para **analise de impacto.**				X		FI
	4	Através do **framework configurável**, são definidos os usuários responsáveis pela aprovação ou rejeição da mudança				X		FI
	5	A ferramenta **notifica via e-mail** quando uma solicitação é **aprovada ou rejeitada**.				X		FI
	PF							FI
	SP1.4-2	**Manter Rastreabilidade Bidirecional entre os Requisitos e o plano de projeto e demais artefatos do projeto**						
	1	Definição de rastreabilidade para frente e para trás **(bidirecional)** entre requisitos.				X		FI
	2	**Há Rastreabilidade para outros artefatos** do projeto através a associação de **arquivos externos** e algumas **ferramentas externas** através de *add-ins* (por exemplo tarefas do MSProject).				X		FI
	3	A ferramenta oferece **matrizes de rastreabilidade** e **diagramas de rastreabilidade**.				X		FI
	4	**Visões customizadas para consultas** específicas sobre os requisitos.				X		FI
	PF							FI
	SP1.5-1	**Identificar Inconsistências entre o plano do projeto e demais artefatos e os requisitos**						
	1	A ferramenta **mostra inconsistências** entre requisitos automaticamente **na matriz de requisitos**.					X	FI
	2	A ferramenta tem recurso para **mostrar inconsistências** entre requisitos e demais artefatos através de verificação manual.		As verificações devem ser customizadas manualmente.		X		LI
	PF							LI
	GG2	**Institucionalizar um processo gerenciado**						
	GP2.1	**Estabelecer e manter uma política organizacional para o planejamento e execução do processo de gerenciamento de requisitos.**						
	1	A ferramenta oferece **componente para criação de *templates* customizáveis** para definição de políticas de gerência de requisitos.					X	LI
	PF							LI
	GP2.2	**Estabelecer e manter o plano para a execução do processo de gerenciamento de requisitos.**						

	1	A ferramenta oferece componente para criação de *templates* customizáveis para definição do plano de execução do processo de gerência de requisitos.					X	LI
	PF							LI
	GP2.3	**Fornecer os recursos adequados para a execução do processo de gerenciamento de requisitos, desenvolvimento de produtos de trabalho e fornecimento dos serviços do processo.**						
	1	O uso da ferramenta CaliberRM provê os recursos através de recursos diretos e *templates* customizáveis para o execução do processo de gerência de requisitos.				X		FI
	PF							FI
	GP2.4	**Atribuir responsabilidades e autoridade para a execução do processo, desenvolvimento dos produtos de trabalho e fornecimento dos serviços do processo de gerenciamento de requisitos.**						
	1	A ferramenta possui recursos para alocação dos recursos às realizações dos requisitos e definição de papéis no módulo de cadastramento de usuários e grupos.				X		FI
	PF							FI
	GP2.5	**Treinar as pessoas para executar e suportar o processo de gerenciamento de requisitos, conforme necessário.**						
	1	A ferramenta fornece **componente para criação de *templates*** customizáveis para registro de treinamentos.					X	LI
	PF							LI
	GP2.6	**Colocar os produtos de trabalho definidos do processo de gerenciamento de requisitos sob os níveis apropriados de gerenciamento de configurações.**						
	1	O CaliberRM possui **versionamento automático** dos requisitos, criação e **gerenciamento de *baselines*** através do ***Baseline Administrator*** e um **framework pré-definido para revisão e aprovação** destas.				X		FI
		A ferramenta possui, também, recursos para a **integração com ferramentas de controle de versão**, como o **CVS** e o StarTeam da Borland.				X		FI
	PF							FI
	GP2.7	**Identificar e envolver os stakeholders relevantes do processo de gerenciamento de requisitos, conforme planejado.**						
	1	Há recurso para **identificação e definição de responsabilidades** dos *stakeholder* no módulo de **cadastramento de usuários**.				X		FI
		Os usuários atribuídos aos requisitos são **notificados por e-mail.**				X		FI
	PF							FI
	GP2.8	**Monitorar e controlar o processo de gerenciamento de requisitos contra o plano de execução do processo e tomar a ação corretiva apropriada.**						
	1	A **monitoração é parte manual** e parte automática, dependendo do artefato alterado, e registrada em documentos baseados em ***templates* customizáveis**.					X	LI
	PF							LI
	GP2.9	**Avaliar objetivamente a aderência do processo de gerenciamento de requisitos contra sua descrição de processo, padrões e procedimentos e tratar as não conformidades.**						
	1	A avaliação é feita de forma manual apoiada por **relatórios de status** e **históricos de alterações** e registrada em ***templates* customizados para relatórios de avaliação**.					X	LI
	PF							LI
	GP2.10	**Revisar as atividades, status e resultados do processo de gerenciamento de requisitos com o nível mais alto nível de gerência e resolver questões.**						

	1	A revisão com a gerência de nível superior é apoiada por **relatórios de status** e **históricos de alterações, métricas sobre alterações e evolução dos requisitos** e registrada em ***templates* customizáveis para relatórios de revisão**.					X	LI
							X	LI
	PF							LI
Resultado geral da avaliação								**S**

Tabela 9 - Planilha de avaliação da ferramenta CaliberRM.

5.3.3 Resultados obtidos

A ferramenta CaliberRM provêm muitos recursos para o gerenciamento de requisitos. Esta ferramenta faz parte de uma suíte de ferramentas para gerenciamento do ciclo de vida do projeto de forma que algumas práticas são suportadas em outras ferramentas. As práticas específicas têm um bom suporte para sua implementação através da ferramenta, porém para as práticas genéricas existem algumas deficiências, pois há uma grande necessidade de customização de *templates*. A maior deficiência foi a falta da definição de um fluxo de trabalho na ferramenta. Desta forma, é necessário o uso de alguns artifícios para que estas práticas sejam alcançadas. O artifício que é mais utilizado, inclusive sendo indicado na documentação da ferramenta é a definição de tarefas de monitoramento e revisão como requisitos do projeto, criando *templates* para a geração da documentação destas atividades.

Desta forma, a avaliação mostrou que a ferramenta CaliberRM está largamente alinhada à PA de Gerência de Requisitos do CMMI, porém há muitos pontos a serem melhorados para uma melhor automatização do processo. Como pode ser observado na planilha, a maioria das práticas podem ser implementadas com o uso de *templates*, o que torna as evidências fracas. Porém cabe uma ressalva: segundo o fabricante, o uso de todas a ferramentas de sua suíte de apoio ao ciclo de vida do projeto dá suporte à todas as áreas de processo do CMMI, automatizando de forma mais consistente os processos.

5.4 AVALIAÇÃO DA FERRAMENTA REQUISITEPRO

5.4.1 Descrição da Ferramenta

O RequisitePro[5] é uma ferramenta para gerência de requisitos desenvolvida pela Rational, de propriedade da IBM, cujo objetivo é ajudar equipes a gerenciar os requisitos durante o ciclo

de vida de um projeto através da promoção da comunicação e colaboração entre os participantes deste projeto. A ferramenta faz parte de uma suíte de ferramentas da Rational para apoio ao ciclo de vida da aplicação, com base no RUP[6] (*Rational Unified Process*), uma metodologia de desenvolvimento de propriedade do fabricante da ferramenta. Desta forma, a ferramenta é apoiada por uma série de *templates*, definidos pelo RUP para apoiar o processo de gerência de requisitos. Além do fornecimento dos *templates*, o fornecedor disponibiliza diversos documento e tutoriais para auxiliar a organização a definir e executar este processo.

A ferramenta utiliza a combinação de um banco de dados de requisitos e o editor de textos Word, da Microsoft, para documentar e manter os requisitos. O banco de dados padrão é o Access, da Microsoft, porém é possível utilizar os bancos de dados DB2 da IBM ou o SQLServer também da Microsoft.

O RequisitePro mantém os requisitos nos documentos associando estes ao banco de dados. Desta forma, é usual definir requisitos diretamente no editor de textos e utilizar um menu especial do RequisitePro (incorporado ao Word) para gerenciar algumas funções da ferramenta. Na base de dados ficam gravados os atributos e demais propriedades do requisito. A descrição textual fica armazenada no próprio documento. Na figura 37 a tela do Word é mostrada, juntamente com o menu que aparece quando editamos o documento a partir da ferramenta RequisitePro.

---

[5] Rational RequisitePro, é um software para gerência de requisitos desenvolvido pela IBM Rational e pode ser baixado, em versão trial (roda por 30 dias), de www.rational.com.

[6] RUP (*Rational Unified Process*) é uma metodologia de desenvolvimento de propriedade da Rational e pode ser baixado, em versão trial, de www.rational .com.

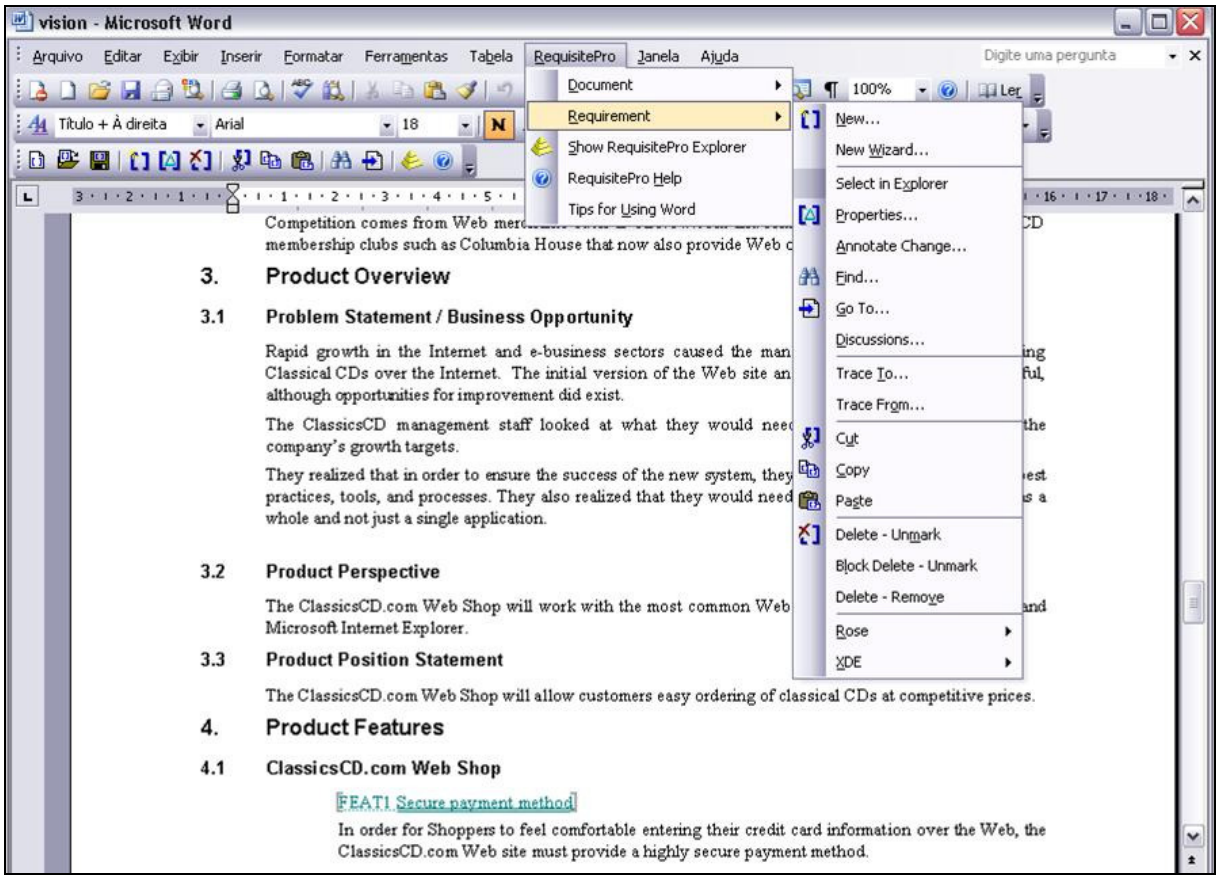


Figura 37 - Interface do Word com integração ao RequisitePro.

Os requisitos são organizados em projetos. Cada projeto pode ter pacotes e dentro destes pacotes estão organizados os requisitos. Para suprir os vários tipos de artefatos que são essenciais a um projeto, o produto oferece alguns tipos de requisitos (funcionais, suplementares) e é possível criar tipos de requisitos customizados para atender outros artefatos que forem necessários.

Desta forma pode-se ter um requisito do tipo "*use case*" (casos de uso), onde é possível customizar atributos para suportar as informações de um caso de uso. Uma característica da ferramenta é guardar varias informações nos documentos, os atributos que foram customizados ficam armazenados nos arquivos, do Word, que documentam os requisitos. Na figura 38 é mostrada a interface do RequisitePro, com um projeto exemplo aberto.

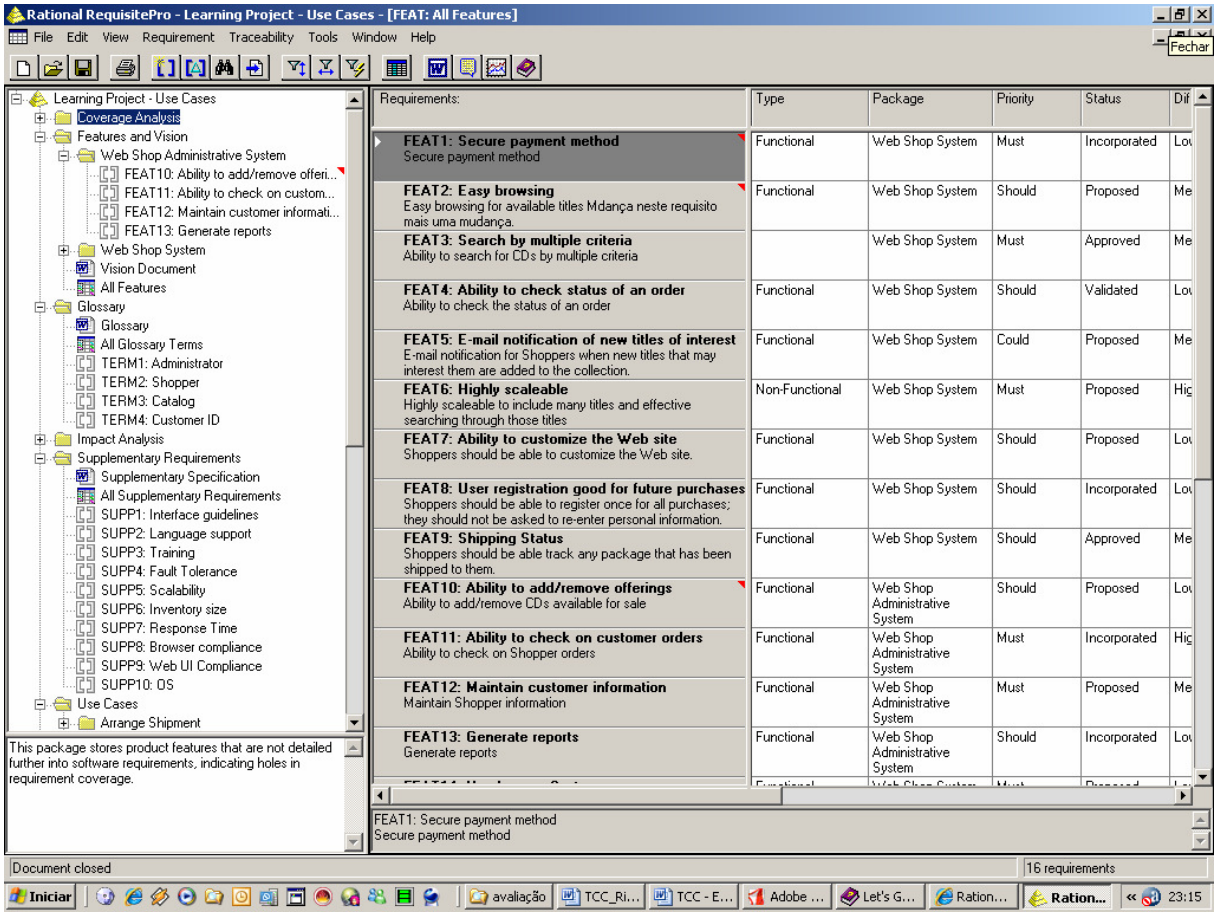


Figura 38 - Interface da ferramenta RequisitePro.

Na figura 38 pode-se ver o projeto, seus pacotes e os requisitos funcionais, identificados pelo prefixo FEAT (de *feature*), os requisitos suplementares, identificados pelo prefixo SUPP (de *suppementary*) e os termos do glossário identificador pelo prefixo TERM. Os documentos aparecem na estrutura do projeto identificados pelo ícone do aplicativo Word ( ).

Um projeto é composto por uma base de dados onde os requisitos ficam armazenados, documentos gerados e informações de rastreamento e gerenciamento. Algumas características estão listadas:

- Tipos de documentos: glossário, visão e "*use cases*" (casos de uso)
- Tipos de requisitos: características, casos de uso, especificações suplementares e demais requisitos customizados.
- Atributos de requisitos: prioridade, status, estabilidade.
- Rastreamento: relacionamentos entre os requisitos – matriz de atributos, matriz de rastreamento e árvore de rastreamento.

Cada requisito do projeto possui controle interno de versão e demais propriedades. Na figura 39 é mostrada a tela de propriedades do requisito.

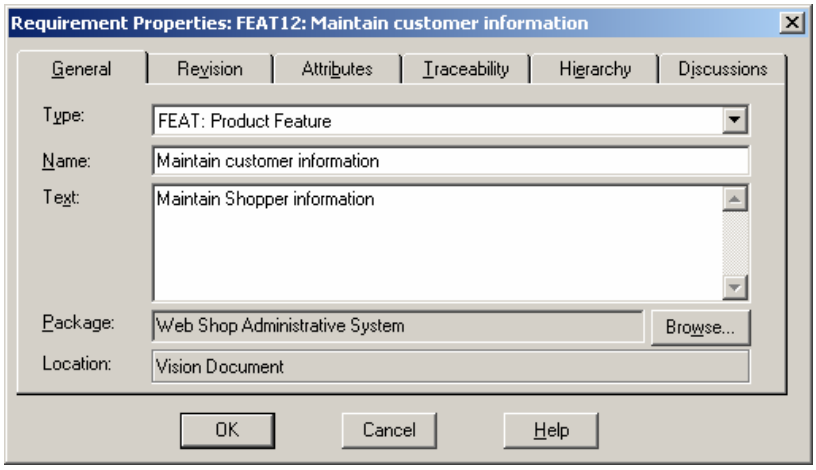


Figura 39 - Propriedades do requisito no RequisitePro.

Também é possivel a criação de usuários e grupos de usuários. Desta forma é possivel definir quem será notificado das mudanças ocorridas nos requisitos. Esta configuração pode ser feita por projeto. Na figura 40 são mostradas as telas de criação de usuário e definição de suas propriedades.

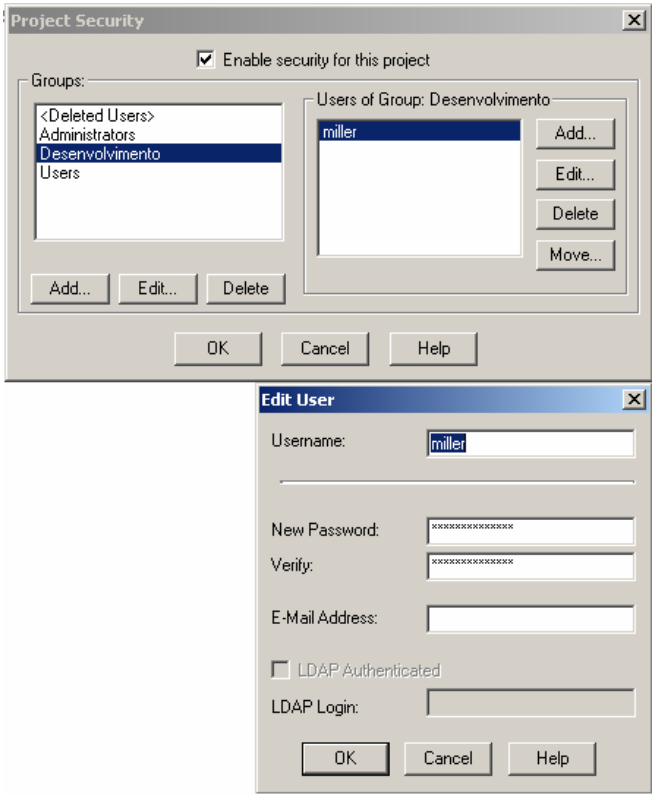


Figura 40 - Propriedades do usuário no RequisitePro.

O RequisitePro notifica os usuários sobre as alterações que são feitas em um requisito. Da mesma forma, possui um recurso chamado “discussão” onde os usuários podem deixar

dúvidas sobre o sobre o requisito para serem respondidas. Na figura 41 é mostrada a tela de de discussão de um requisito.

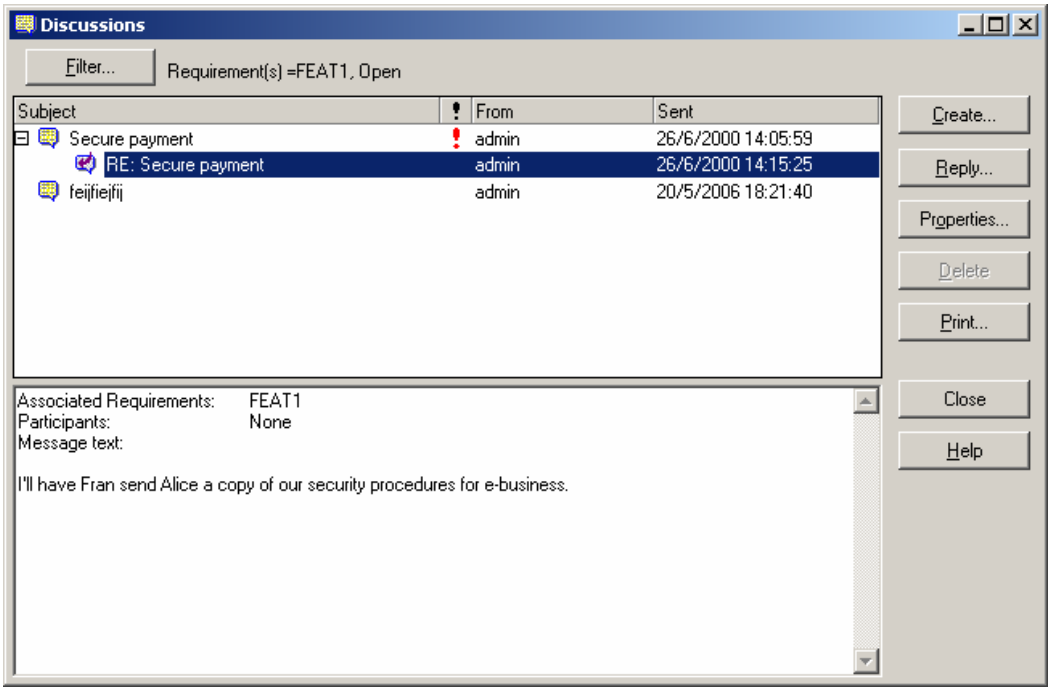


Figura 41 - Recursos de discussões do RequisitePro.

Também é possível fazer a manutenção de *baselines* do projeto. Estas *baselines* incluem todos as artefatos do projeto e podem ser configuradas para ter a necessidade de aprovação por usuários com permissão para este recurso. As *baselines* são gerenciadas por um componente chamado *Rational Baseline Manager*. A cada fechamento de uma *baseline*, o fabricante sugere a criação de um novo projeto para suprir as alterações para um próximo *release* do produto. Observou-se que as *baselines* podem ser exportadas em arquivos XML. A figura 41 mostra a interface do *Rational Baseline Manager*.

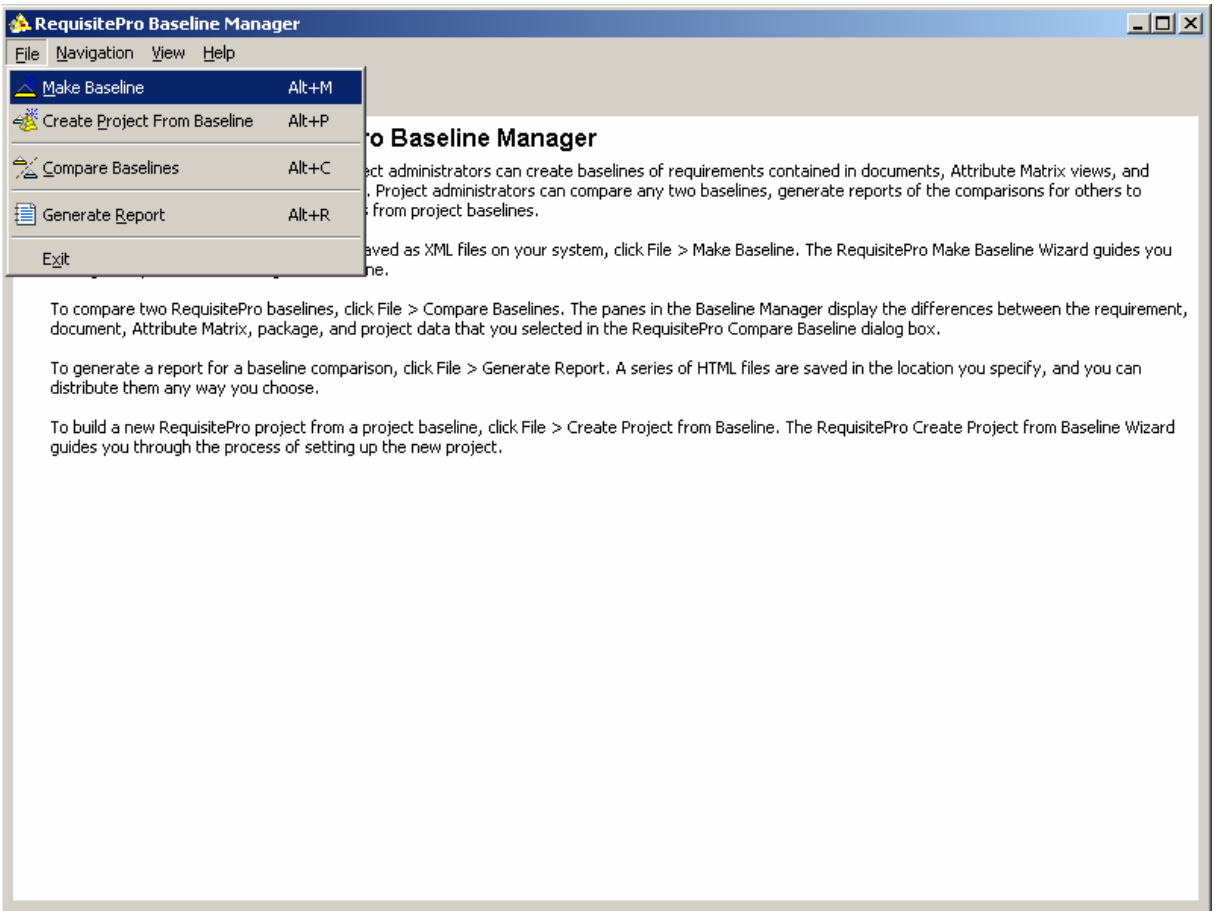

RequisitePro Baseline Manager
File Navigation View Help
Make Baseline Alt+M
Create Project From Baseline Alt+P
Compare Baselines Alt+C
Generate Report Alt+R
Exit
ro Baseline Manager
ct administrators can create baselines of requirements contained in documents, Attribute Matrix views, and
. Project administrators can compare any two baselines, generate reports of the comparisons for others to
from project baselines.
aved as XML files on your system, click File > Make Baseline. The RequisitePro Make Baseline Wizard guides you
ne.
To compare two RequisitePro baselines, click File > Compare Baselines. The panes in the Baseline Manager display the differences between the requirement, document, Attribute Matrix, package, and project data that you selected in the RequisitePro Compare Baseline dialog box.
To generate a report for a baseline comparison, click File > Generate Report. A series of HTML files are saved in the location you specify, and you can distribute them any way you choose.
To build a new RequisitePro project from a project baseline, click File > Create Project from Baseline. The RequisitePro Create Project from Baseline Wizard guides you through the process of setting up the new project.

Figura 42 - *Baseline Manager* do RequisitePro.

O fabricante sugere o uso do Rational ClearCase para gerenciamento completo de *baselines*. Para analisar uma *baseline*, pode-se utilizar a opção *Compare baselines* que gera um relatório das diferenças entre as *baselines* do projeto.

A ferramenta possui arquitetura de forma que seja possível que os requisitos de software sejam ligados a uma variedade de artefatos em todo o ciclo de vida. Esta ligação pode ser direta a outros requisitos ou utilisando-se de drivers de acesso às artefatos em outras aplicações (por exemplo, tarefas de um projeto no MSProject). Há ainda a possibilidade de vincular documentos externos. Existem dois tipos de ligações, uma indica que atefatos dependem do requisito e de quais artefatos o requisito é dependente. A figura 43 mostra a tela onde é definida a rastreabilidade de requisito.

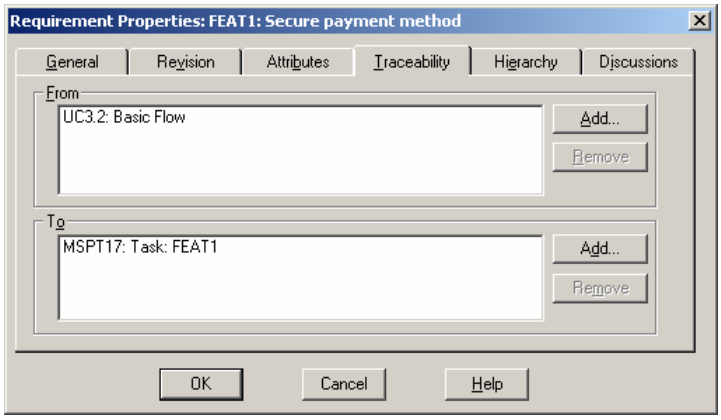


Figura 43 - Definição da rastreabilidade do requisito.

Para consultar os requisitos, a ferramenta disponibiliza visões. Desta forma, requisitos, seus atributos e relacionamentos podem ser criados, visualizados e gerenciados em visões. As visões podem ser criadas pela ordenação e filtragem de requisitos e seus atributos, podem ser criadas na forma de matrizes ou árvore e podem ser usadas para filtrar atributos ou exibir relacionamentos entre requisitos. Existem três tipos diferentes de visões: Matriz de Atributos (*Attribute Matrix*), Matriz de Rastreabilidade (*Traceability Matrix*) e Árvore de Rastreabilidade (*Traceability Tree*). As figuras 44, 45 e 46 mostram as três possibilidades de visões.

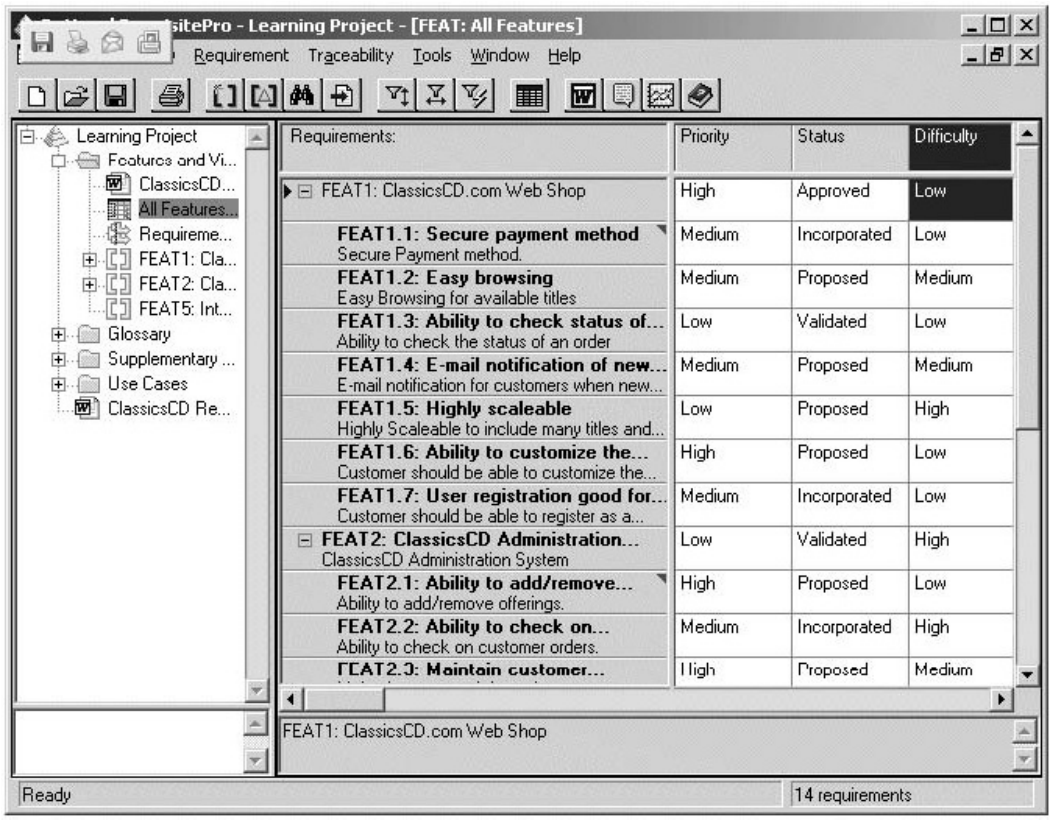


Figura 44 - Matriz de atributos dos requisitos.

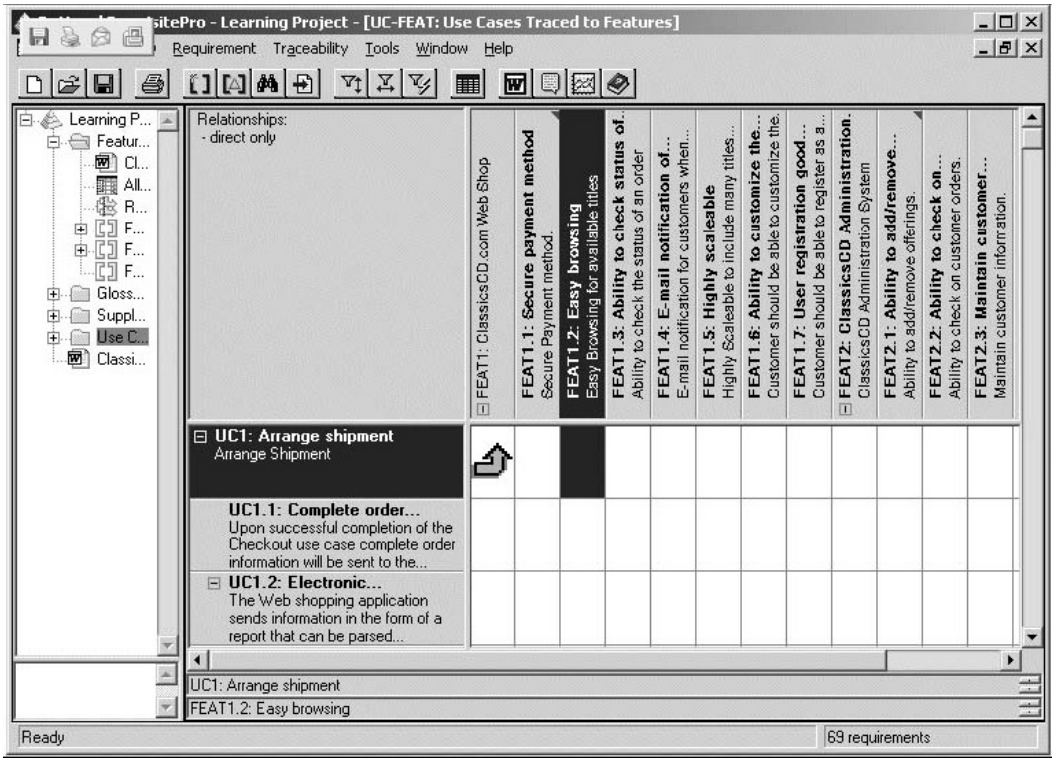

sitePro - Learning Project - [UC-FEAT: Use Cases Traced to Features]
Requirement Traceability Tools Window Help
Relationships:
- direct only
FEAT1: ClassicsCD.com Web Shop
FEAT1.1: Secure payment method
FEAT1.2: Easy browsing
FEAT1.3: Ability to check status of...
FEAT1.4: E-mail notification of...
FEAT1.5: Highly scaleable
FEAT1.6: Ability to customize the...
FEAT1.7: User registration good...
FEAT2: ClassicsCD Administration.
FEAT2.1: Ability to add/remove...
FEAT2.2: Ability to check on...
FEAT2.3: Maintain customer...
UC1: Arrange shipment
Arrange Shipment
UC1.1: Complete order...
Upon successful completion of the Checkout use case complete order information will be sent to the...
UC1.2: Electronic...
The Web shopping application sends information in the form of a report that can be parsed...
UC1: Arrange shipment
FEAT1.2: Easy browsing
Ready
69 requirements

Figura 45 - Matriz de rastreabilidade dos requisitos.

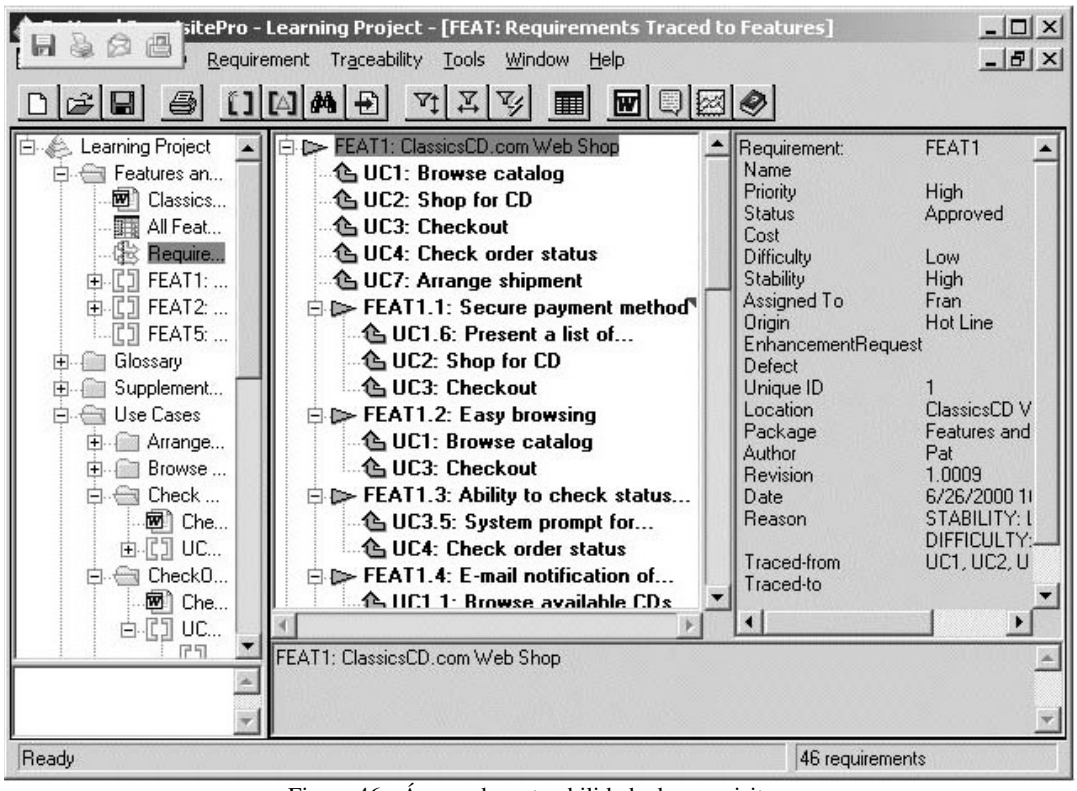


Figura 46 - Árvore de rastreabilidade dos requisitos.

O RequisitePro possui interface não muito intuitiva e fácil de utilizar e possui muitas funcionalidades de suporte à decisão e análise de impacto utilizando-se das visões

disponíveis, além de também utilizar recursos de correio eletrônico permitindo que todos os envolvidos no projeto colaborem entre si durante o ciclo de vida do projeto. O fabricante da ferramenta indica que a ferramenta é aderente ao nível dois do CMMI desde que seja utilizada em conjunto com outras ferramentas da Suíte da Rational. A orientação do fabricante é o uso do RequisitePro em conjunto com as outras ferramentas para que se tenha um alinhamento completo com o CMMI utilizando-se de uma gerência de configuração e gerência de mudanças mais eficaz. E interessante ressaltar que o fabricante não deixa claro que é possível atingir o nível dois com o uso da ferramenta RequisitePro sem o uso das demais ferramentas do mesmo fabricante. Caso a empresa deseje utilizar o RequisitePro associado à ferramentas de outros fabricantes, ficando a cargo da empresa customizar o uso e a integração da aplicação de gerência de requisitos com os demais aplicativos utilizados.

5.4.2 Aplicação da avaliação

A avaliação foi feita sobre o Rational RequisitePro versão 2003.06.15.734.000, baixado de www.rational.com em abril de 2005. O processo de instalação da ferramenta não será colocado neste trabalho em função de ser um processo intuitivo e bem documentado pelo fabricante. Para a avaliação foi utilizado o projeto exemplo que vem pré-instalado no banco de dados da aplicação.

Como referência para aprendizado da ferramenta utilizou-se a documentação instalada junto com o produto, juntamente com tutoriais também instalados com o produto. Esta documentação mostrou-se completa, suprindo todas as dúvidas e possibilidades de implementações necessárias para a execução desta avaliação.

A tabela 10 mostra os resultados da avaliação desta ferramenta e registra as evidências que foram encontradas para suprir as práticas indicadas na PA Gerência de Requisitos do CMMI.

Gerenciamento de Requisitos - REQM

Ferramenta: Rational RequisitePro

Status		Praticas / Anotações	Documento(s)	Comentários		Direta	Indireta	Pontuação
	SG1	**Os requisitos são gerenciados e as inconsistências com o plano de projeto e produtos de trabalho são identificadas**						
	SP1.1-1	**Obter um entendimento dos Requisitos**						
	1	A ferramenta fornece um **template** para documento de **Especificação de Requisitos do Sistema** onde está previsto uma seção para **definição dos fornecedores de**	*Use case* Specification, Supplementary				X	LI

		requisitos e **critérios para aprovação dos mesmos**.	Requirements Specification					
	3	A ferramenta indica o recurso de **discussão** como ferramenta para **obter um entendimento comum do significado dos requisitos.**					X	LI
	PF							LI
	SP1.2-2	**Obter Comprometimento dos participantes do projeto com Requisitos**						
	1	A ferramenta indica o **recurso de discussão** como ferramenta para obter o comprometimento com os requisitos.					X	LI
	PF							LI
	SP1.3-1	**Gerenciar Mudanças de Requisitos a medida que elas ocorram durante o projeto**						
	1	Qualquer inclusão, alteração ou exclusão é **notificada, por e-mail** ao gerente do projeto.	Stakeholder Request Document			X		FI
	2	Para **análise de impacto**, a ferramenta fornece **uma matriz de rastreabilidade ou na árvore de rastreabilidade**.				X		FI
	4	Os requisitos possuem um **status** que indica se o mesmo foi **aprovado o rejeitado**.				X		FI
	5	**Caso a mudança seja aprovada ou rejeitada**, os solicitantes e usuários configurados **recebem a notificação, por e-mail**.				X		FI
	PF							FI
	SP1.4-2	**Manter Rastreabilidade Bidirecional entre os Requisitos e o plano de projeto e demais artefatos do projeto**						
	1	A ferramenta possui recursos para definir **rastreabilidade bidirecional** com outros **requisitos**, com **arquivos em disco (para artefatos externos)**, **outros softwares** da Rational, com os aplicativos Word e Excel e com o MS Project.				X		FI
	3	Como **ferramentas de rastreabilidade**, a ferramenta oferece **matrizes de rastreabilidade** ou **árvores hierárquicas de rastreabilidade.**				X		FI
	4	A ferramenta fornece **relatórios de rastreabilidade** para verificação das associações de **rastreabilidade para frente e para trás.**				X		FI
	PF							FI
	SP1.5-1	**Identificar Inconsistências entre o plano do projeto e demais artefatos e os requisitos**						
	1	A ferramenta **mostra inconsistências** entre requisitos automaticamente **na matriz de rastreabilidade de requisitos**.					X	LI
	2	A ferramenta tem recurso para **mostrar inconsistências** entre requisitos e demais artefatos através de verificação manual.		Manualmente, o revisor marca as inconsistências para que os responsáveis efetuem a ação corretiva.			X	LI
	PF							LI
	GG2	**Institucionalizar um processo gerenciado**						
	GP2.1	**Estabelecer e manter uma política organizacional para o planejamento e execução do processo de gerenciamento de requisitos.**						
	1	A ferramenta oferece **componente para criação de *templates*** onde há ma seção para definição de políticas de gerência de requisitos.	Requirements Management Plan				X	LI
	PF							LI
	GP2.2	**Estabelecer e manter o plano para a execução do processo de gerenciamento de requisitos.**						
	1	A ferramenta oferece **componente para criação de *templates*** onde há ma seção para o plano de execução do processo de gerenciamento de requisitos é cria ao projeto	Requirements Management Plan	.			X	LI

		no formato do editor de textos Word.						
	PF							**LI**
	GP2.3	**Fornecer os recursos adequados para a execução do processo de gerenciamento de requisitos, desenvolvimento de produtos de trabalho e fornecimento dos serviços do processo.**						
	1	**O uso da ferramenta RequisitePro provê os recursos**, através do fornecimento de ***templates*** e **ferramentas** diretas para a execução do processo de Gerência de requisitos.				**X**		**FI**
	PF							**FI**
	GP2.4	**Atribuir responsabilidades e autoridade para a execução do processo, desenvolvimento dos produtos de trabalho e fornecimento dos serviços do processo de gerenciamento de requisitos.**						
	1	Esta prática é atendida pelo uso do **template** do documento de **Especificação de Requisitos do Sistema** onde há uma seção própria para **definição dos papéis e responsabilidades** no processo de gerência de requisitos.					**X**	**LI**
	PF							**LI**
	GP2.5	**Treinar as pessoas para executar e suportar o processo de gerenciamento de requisitos, conforme necessário.**						
	1	A ferramenta fornece ***templates*** **customizáveis** para registro dos treinamentos dos stakeholders.					**X**	**LI**
	PF							**LI**
	GP2.6	**Colocar os produtos de trabalho definidos do processo de gerenciamento de requisitos sob os níveis apropriados de gerenciamento de configurações.**						
	1	Os **requisitos possuem versionamento** interno fornecido pela ferramenta. É possível criar *baselines* com o componente ***Baseline Manager***, atingindo todos os artefatos e documentos associados ao projeto.					**X**	**LI**
	PF							**LI**
	GP2.7	**Identificar e envolver os stakeholders relevantes do processo de gerenciamento de requisitos, conforme planejado.**						
	1	A ferramenta fornece o ***templates*** para o documento de **Especificação de Requisitos do Sistema** e o documento de **Visão do sistema** que possuem sessões para identificação dos participantes do projeto.					**X**	**LI**
	PF							**LI**
	GP2.8	**Monitorar e controlar o processo de gerenciamento de requisitos contra o plano de execução do processo e tomar a ação corretiva apropriada.**						
	1	A **monitoração é parte manual** e parte automática (notificação por e-mail), dependendo do artefato alterado, e registrada em documentos baseados em ***templates*** **customizáveis**.		A monitoração é manual e de responsabilidade do usuário. A ferramenta indica porém não dá suporte direto à este recurso.			**X**	**LI**
	PF							**LI**
	GP2.9	**Avaliar objetivamente a aderência do processo de gerenciamento de requisitos contra sua descrição de processo, padrões e procedimentos e tratar as não conformidades.**						
	1	A avaliação é feita de forma manual apoiada por **relatórios de status** e **históricos de alterações** e registrada em ***templates*** **customizados para relatórios de avaliação**.					**X**	**LI**
	PF							**LI**
	GP2.10	**Revisar as atividades, status e resultados do processo de gerenciamento de requisitos com o nível mais alto nível de gerência e resolver questões.**						
	1	A revisão com a gerência de nível superior é apoiada por **relatórios de status**, **relatório de histórico de alterações, comparações entre baselines** e registrada em ***templates***					**X**	**LI**

		de documentos de revisão.						
	PF							LI
Resultado geral da avaliação								S

Tabela 10 - Planilha de avaliação da ferramenta Rational RequisitePro.

5.4.3 Resultados obtidos

A ferramenta Rational RequisitePro, da mesma forma que o CaliberRM, provêm muitos recursos para o gerenciamento de requisitos. Esta ferramenta faz parte de uma suíte de ferramentas para engenharia de software da Rational que dá suporte ao RUP (*Rational Unified Process*) que é um modelo de processo alinhado ao CMMI. Desta forma há vários *templates* disponíveis para apoio ao processo de gerência de requisitos.

Individualmente observou-se a falta de alguns recursos diretos para apoio a algumas práticas e uma fragilidade na maneira de gerenciar os documentos do projeto. A ferramenta indica o uso de *templates* para armazenagem de várias informações, como por exemplo a definição dos papéis e responsabilidades dos *stakeholders*, registro de reuniões etc. Porém, como esta informação é textual, não há como garantir que o *template* foi utilizado de forma correta. Outro aspecto é a falta de um *framework* para dar suporte ao processo de gerência de requisitos. Segundo a documentação do fabricante há a necessidade do uso da ferramenta Rational ClearQuest para que o processo de controle de mudanças tenha um suporte mais automatizado. O fabricante recomenda o uso da ferramenta Rational Process Workbench em conjunto com o Rational RequisitePro para que seja possível a definição e execução de um fluxo de trabalho para o processo de gerência de requisitos.

Da mesma forma, a geração de *baselines* de um projeto pode ser feita livremente não havendo a necessidade de qualquer aprovação ou verificação por parte dos usuários. Segundo documentação do fabricante, para um suporte completo à gerência de configuração, inclusive com a implementação de um *framework*, há a necessidade do uso da ferramenta Rational ClearCase. Outro aspecto foi o fato que a ferramenta gera as *baselines* pela da exportação do projeto em arquivos XML, ou seja, o RequisitePro não armazena suas *baselines* no banco de dados de requisitos. O processo de recuperação destas *baselines* também foi considerado frágil, uma vez que é necessária a criação de outro projeto na ferramenta e a partir daí efetuar a importação dos arquivos XML gerados para a *baseline*.

Mesmo com alguns pontos fracos, a avaliação mostrou que a ferramenta Rational RequisitePro está alinhada à PA de Gerência de Requisitos do CMMI. Como pode ser

observado na planilha, a maioria das práticas podem ser implementadas com o uso de *templates*, o que evidenciou a falta de recursos que forneçam apoio direto à implementação destas. Porém cabe uma ressalva: segundo o fabricante, o uso de todas as ferramentas de sua suíte dá suporte direto à todas as áreas de processo do CMMI, considerando como referência o nível dois de maturidade. Observou-se que, em ralação à ferramenta CaliberRM, o RequisitePro mostrou-se menos completo quando utilizado individualmente uma vez que sua dependência das demais ferramentas da Rational é alta.

Para fins de comparação entre as ferramentas comerciais, foi criada uma tabela mostrando as diferenças, em relação à PA Gerência de Requisitos, entra as ferramentas Borland CaliberRM e Rational RequisitePro. Foi verificado que, quando utilizada de forma individual, o CaliberRM mostrou-se mais completo que o RequisitePro. A tabela 11 faz uma comparação entre as ferramentas analisadas com relação às práticas sugeridas pelo modelo CMMI.

	Praticas da PA Gerência de Requisitos	**Pontuação**	
		CaliberRM	**RequisitePro**
SG1	**Os requisitos são gerenciados e as inconsistências com o plano de projeto e produtos de trabalho são identificadas**	**S**	**S**
SP1.1-1	**Obter um entendimento dos Requisitos**	**LI**	**LI**
SP1.2-2	**Obter Comprometimento dos participantes do projeto com Requisitos**	**FI**	**LI**
SP1.3-1	**Gerenciar Mudanças de Requisitos a medida que elas ocorram durante o projeto**	**FI**	**FI**
SP1.4-2	**Manter Rastreabilidade Bidirecional entre os Requisitos e o plano de projeto e demais artefatos do projeto**	**FI**	**FI**
SP1.5-1	**Identificar Inconsistências entre o plano do projeto e demais artefatos e os requisitos**	**LI**	**LI**
GG2	**Institucionalizar um processo gerenciado**	**S**	**S**
GP2.1	**Estabelecer e manter uma política organizacional para o planejamento e execução do processo de gerenciamento de requisitos.**	**LI**	**LI**
GP2.2	**Estabelecer e manter o plano para a execução do processo de gerenciamento de requisitos.**	**LI**	**LI**
GP2.3	**Fornecer os recursos adequados para a execução do processo de gerenciamento de requisitos, desenvolvimento de produtos de trabalho e fornecimento dos serviços do processo.**	**FI**	**FI**
GP2.4	**Atribuir responsabilidades e autoridade para a execução do processo, desenvolvimento dos produtos de trabalho e fornecimento dos serviços do processo de gerenciamento de requisitos.**	**FI**	**LI**
GP2.5	**Treinar as pessoas para executar e suportar o processo de gerenciamento de requisitos, conforme necessário.**	**LI**	**LI**
GP2.6	**Colocar os produtos de trabalho definidos do processo de gerenciamento de requisitos sob os níveis apropriados de gerenciamento de configurações.**	**FI**	**LI**
GP2.7	**Identificar e envolver os stakeholders relevantes do processo de gerenciamento de requisitos, conforme planejado.**	**FI**	**LI**

GP2.8	**Monitorar e controlar o processo de gerenciamento de requisitos contra o plano de execução do processo e tomar a ação corretiva apropriada.**	**LI**	**LI**
GP2.9	**Avaliar objetivamente a aderência do processo de gerenciamento de requisitos contra sua descrição de processo, padrões e procedimentos e tratar as não conformidades.**	**LI**	**LI**
GP2.10	**Revisar as atividades, status e resultados do processo de gerenciamento de requisitos com o nível mais alto nível de gerência e resolver questões.**	**LI**	**LI**

Tabela 11 - Comparativo entre as ferramentas CaliberRM e RequisitePro.

Observa-se, pelos dados comparativos mostrados na planilha de avaliação, que ambas as ferramentas, apóiam de forma satisfatória a PA Gerência de Requisitos, mesmo não tendo sido desenvolvidas para este fim. O alto grau de customização e interconexão com outras ferramentas do mercado reflete que ambas procuram atingir sua finalidade de forma independente da metodologia de desenvolvimento adotada ou do modelo de processo utilizado pelas organizações que as adotam. Desta forma, fica claro que estas ferramentas comerciais têm alto grau de maturidade e aplicabilidade, o que reflete seu reconhecimento pelo mercado.

5.5 AVALIAÇÃO DA FERRAMENTA PLACES

5.5.1 Descrição da Ferramenta

PLACES é uma plataforma colaborativa desenvolvida para apoiar a gerência de projetos de software, flexibilizando a construção de ferramentas que venham oferecer suportar diferentes processos da Engenharia de Software. (MORETTO, 2005). Esta plataforma é uma evolução da ferramenta colaborativa ARCase, desenvolvida inicialmente para suportar apenas a fase de concepção de software. A proposta do PLACES é ser um ambiente comum para a construção incremental de diferentes ferramentas onde os recursos colaborativos já estão disponíveis para esta plataforma. Desta forma, a construção das ferramentas é facilitada e focada nas boas práticas associadas com os processos que elas pretendem trabalhar. No trabalho de Moretto (2005), foi desenvolvida uma ferramenta para gerência de projetos, incluindo as áreas de planejamento e monitoração de projetos. Entretanto, foi desenvolvida toda uma infra-estrutura de segurança e de colaboração (ferramentas para reuniões virtuais, fóruns de discussão e outros recursos). No trabalho de SOUZA (2005), foi desenvolvido um módulo para documentação de casos de uso com um módulo de estimativa de tamanho baseado em pontos

de caso de uso. A figura 47 apresenta uma tela da plataforma PLACES, onde é possível selecionar um dos projetos em andamento.

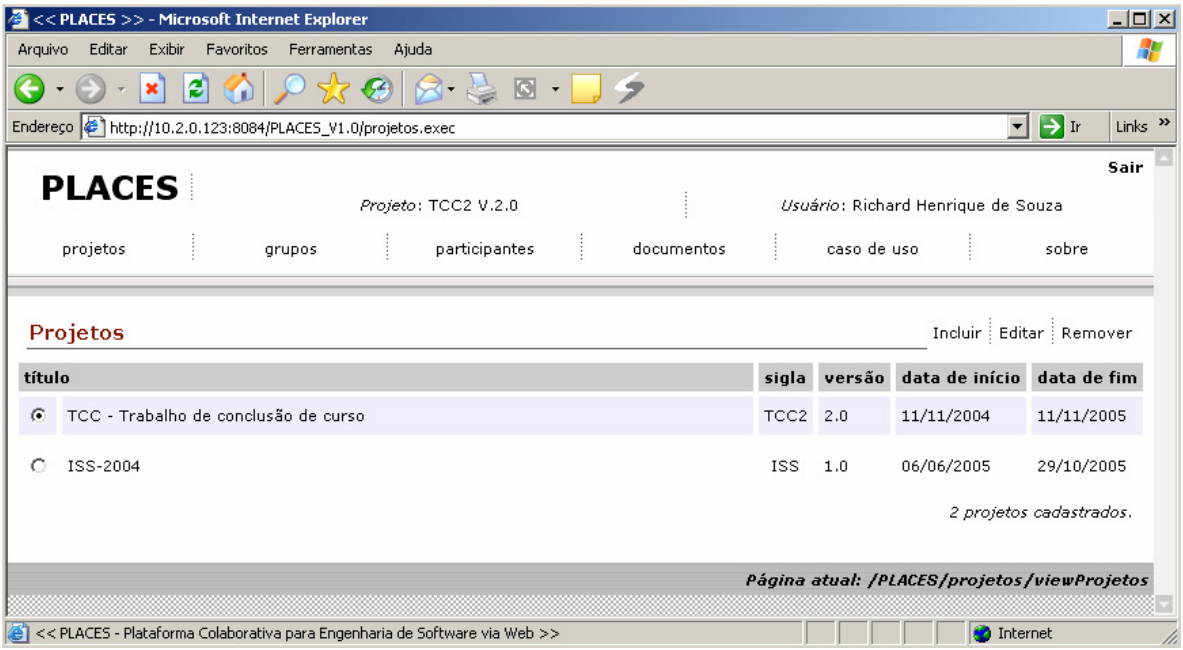


Figura 47 - Interface da plataforma PLACES.

A figura 48 mostra o detalhamento das tarefas do projeto.

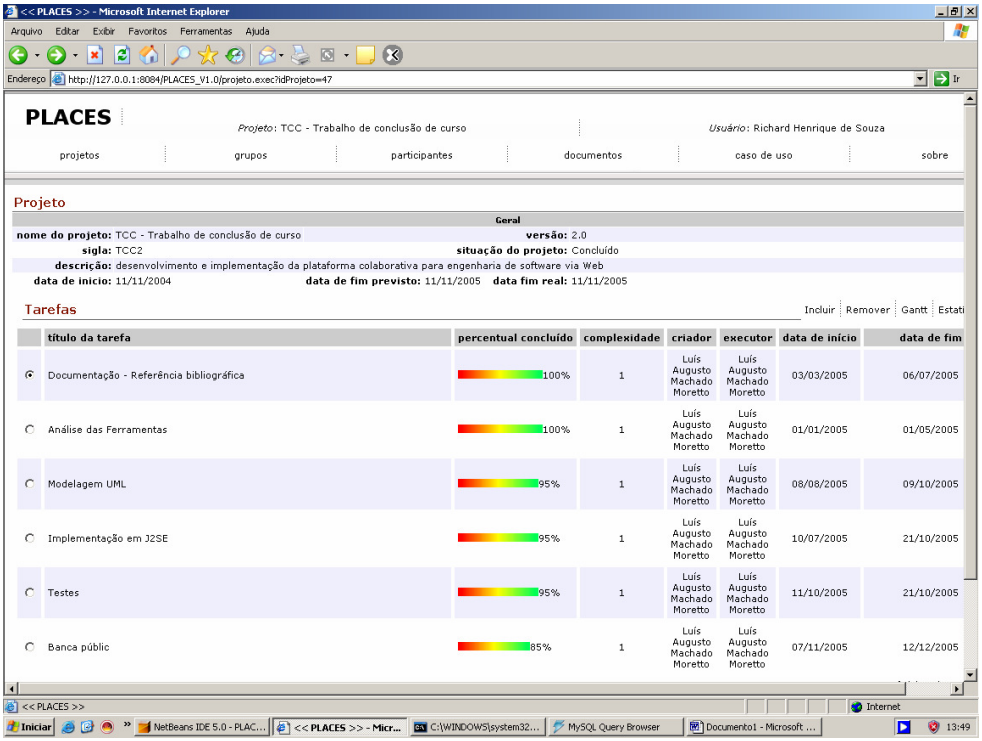

<< PLACES >> - Microsoft Internet Explorer
Endereço http://127.0.0.1:8084/PLACES_V1.0/projeto.exec?idProjeto=47
PLACES
Projeto: TCC - Trabalho de conclusão de curso
Usuário: Richard Henrique de Souza
projetos grupos participantes documentos caso de uso sobre
Projeto
Geral
nome do projeto: TCC - Trabalho de conclusão de curso
versão: 2.0
sigla: TCC2
situação do projeto: Concluído
descrição: desenvolvimento e implementação da plataforma colaborativa para engenharia de software via Web
data de início: 11/11/2004
data de fim previsto: 11/11/2005
data fim real: 11/11/2005
Tarefas
Incluir Remover Gantt
título da tarefa percentual concluído complexidade criador executor data de início data de fim
Documentação - Referência bibliográfica 100% 1 Luís Augusto Machado Moretto 03/03/2005 06/07/2005
Análise das Ferramentas 100% 1 01/01/2005 01/05/2005
Modelagem UML 95% 1 08/08/2005 09/10/2005
Implementação em J2SE 95% 1 10/07/2005 21/10/2005
Testes 95% 1 11/10/2005 21/10/2005
Banca públic 85% 1 07/11/2005 12/12/2005
Internet

Figura 48 - Tarefas do projeto no PLACES.

Neste sentido, o ambiente ARCase passou a ser uma das ferramentas da plataforma PLACES, dando ênfase à área de desenvolvimento de requisitos. Um dos objetivos deste trabalho é a adaptação da ferramenta para dar suporte ao processo de gerência de requisitos. A filosofia inicial do ambiente ARCase continua preservada nesta nova versão, onde a ênfase não está no

desenvolvimento de diagramas, uma vez que ela não possui recursos para a construção de diagramas. O foco continua sendo a manipulação eficiente de documentos, usualmente deixada em segundo plano por ferramentas gráficas. Utilizando os recursos disponibilizados pela plataforma PLACES, a ferramenta atua como um banco de informações compartilhadas entre os participantes do projeto, por meio de um ambiente colaborativo. O caso de uso é o artefato central da ferramenta que permite a integração da concepção. A figura 49 mostra a tela de cadastramento do artefato caso de uso.

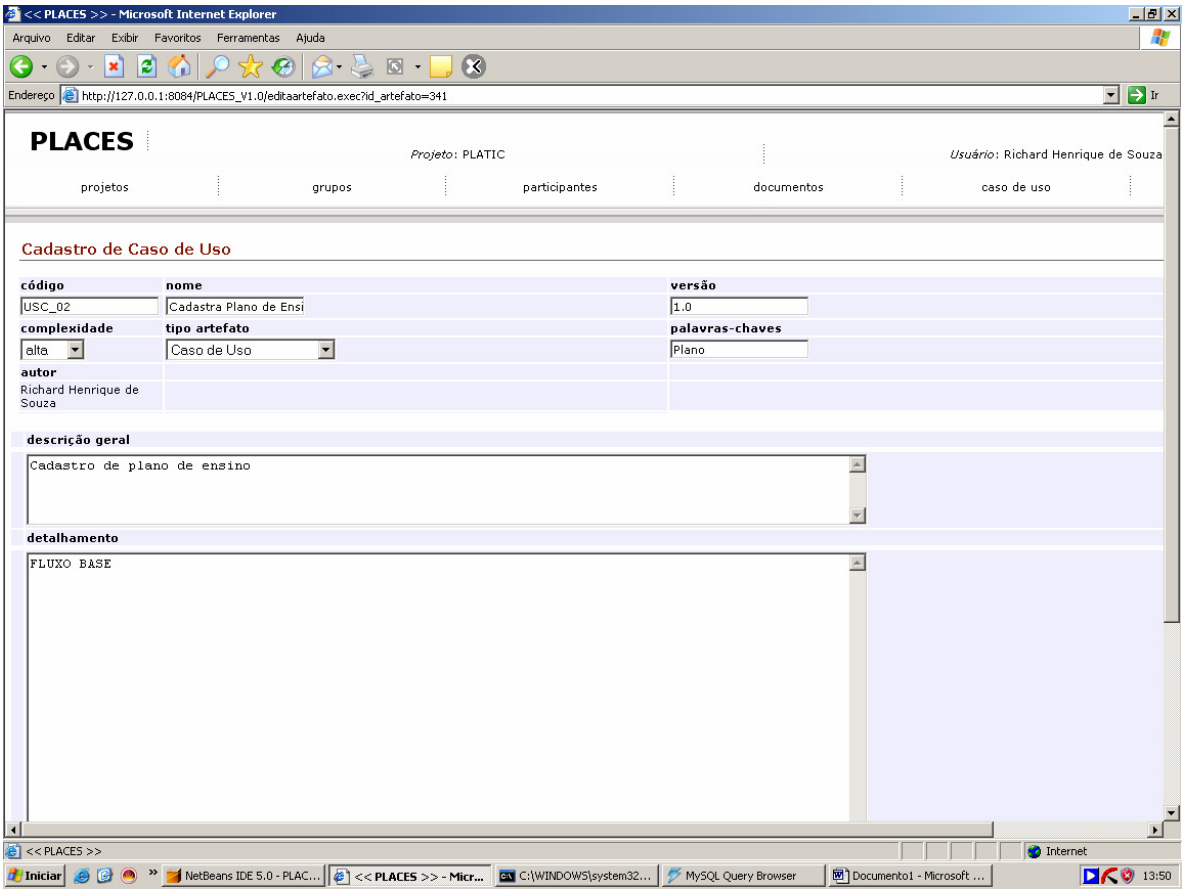


Figura 49 - Cadastramento de casos de uso no PLACES.

Apesar de ser possível o cadastramento de casos de uso, não há recursos diretos para cadastramento de requisitos de forma direta no PLACES, sendo que esta deficiência será o foco da avaliação, buscando identificar quais recursos podem ser reaproveitados no desenvolvimento do módulo de gerência de requisitos.

5.5.2 Aplicação da avaliação

A avaliação foi feita sobre a plataforma PLACES na versão 1.0.1.1 obtida no LQPS/Univali em maio de 2006. O processo de instalação da ferramenta será colocado como anexo a este trabalho em sua segunda etapa (TCC dois) em função de ser um processo extenso. Para a avaliação foi utilizado o projeto exemplo disponível no banco de dados (em sua versão 1.2) disponibilizado para a aplicação.

Como referência para aprendizado da ferramenta utilizou-se os trabalhos de conclusão de Moretto (2005) e Souza (2005) que desenvolveram a ferramenta. Esta documentação, devido ao seu teor acadêmico, mostrou-se incompleta, no ponto de vista de usuário, não suprindo todas as dúvidas e possibilidades de implementações necessárias para a execução desta avaliação, sendo necessário consultar os criadores da ferramenta para obter um entendimento mais detalhado.

A tabela 12 mostra os resultados da avaliação desta ferramenta e registra as evidências que foram encontradas para suprir as práticas indicadas na PA Gerência de Requisitos do CMMI. Da mesma forma como foi feito na avaliação do CaliberRM, a coluna "Documentos" foi deixada na planilha para que não se tenha uma planilha específica para a avaliação de cada ferramenta, apesar do PLACES não fornecer *templates* para geração de documentos.

Gerenciamento de Requisitos - REQM

Ferramenta: PLACES

Status		Praticas / Anotações	Documento(s)	Comentários		Direta	Indireta	Pontuação
	SG1	Os requisitos são gerenciados e as inconsistências com o plano de projeto e produtos de trabalho são identificadas						
	SP1.1-1	Obter um entendimento dos Requisitos						
-	1	A ferramenta oferece cadastramento de casos de uso					X	PI
	PF							PI
	SP1.2-2	Obter Comprometimento dos participantes do projeto com Requisitos						
	PF							NI
	SP1.3-1	Gerenciar Mudanças de Requisitos a medida que elas ocorram durante o projeto						
	PF							NI
	SP1.4-2	Manter Rastreabilidade Bidirecional entre os Requisitos e o plano de projeto e demais artefatos do projeto						

	PF							NI
	SP1.5-1	Identificar Inconsistências entre o plano do projeto e demais artefatos e os requisitos						
	PF							NI
	GG2	Institucionalizar um processo gerenciado						
	GP2.1	Estabelecer e manter uma política organizacional para o planejamento e execução do processo de gerenciamento de requisitos.						
	PF							NI
	GP2.2	Estabelecer e manter o plano para a execução do processo de gerenciamento de requisitos.						
	PF							NI
	GP2.3	Fornecer os recursos adequados para a execução do processo de gerenciamento de requisitos, desenvolvimento de produtos de trabalho e fornecimento dos serviços do processo.						
		A ferramenta fornece recursos parciais para execução do processo, atendendo parcialmente o processo de entendimento comum sobre os requisitos.					X	PI
	PF							PI
	GP2.4	Atribuir responsabilidades e autoridade para a execução do processo, desenvolvimento dos produtos de trabalho e fornecimento dos serviços do processo de gerenciamento de requisitos.						
	PF							NI
	GP2.5	Treinar as pessoas para executar e suportar o processo de gerenciamento de requisitos, conforme necessário.						
	PF							NI
	GP2.6	Colocar os produtos de trabalho definidos do processo de gerenciamento de requisitos sob os níveis apropriados de gerenciamento de configurações.						
	PF							NI
	GP2.7	Identificar e envolver os stakeholders relevantes do processo de gerenciamento de requisitos, conforme planejado.						
	1	Os integrantes do projeto são identificados através de **cadastro próprio**.		Falta definição de papéis para o envolvimento dos participantes			X	PI
	PF							PI
	GP2.8	Monitorar e controlar o processo de gerenciamento de requisitos contra o plano de execução do processo e tomar a ação corretiva apropriada.						
	PF							NI
	GP2.9	Avaliar objetivamente a aderência do processo de gerenciamento de requisitos contra sua descrição de processo, padrões e procedimentos e tratar as não conformidades.						

	PF							NI
	GP2.10	Revisar as atividades, status e resultados do processo de gerenciamento de requisitos com o nível mais alto nível de gerência e resolver questões.						
	PF							NI
Resultado geral da avaliação								I

Tabela 12 - Planilha de avaliação da ferramenta PLACES.

5.5.3 Resultados obtidos

A ferramenta PLACES provê poucos recursos para o gerenciamento de requisitos, uma vez que esta ferramenta não foi desenvolvida para este fim e o que há é um módulo para cadastramento de casos de uso enfocando a definição de medidas de estimativa baseadas em pontos por caso de uso. Outro aspecto foi a falta de maturidade tendo como referência as ferramentas comerciais avaliadas.

Desta forma, a avaliação mostrou que a ferramenta PLACES não está alinhada à PA de Gerência de Requisitos do CMMI e há muitos pontos a serem melhorados para apoiar este processo. Como pode ser identificado na planilha, a maioria das práticas não são atendidas diretamente e a ferramenta também não oferece *templates* para customização do processo. Porém cabe uma ressalva: este resultado era esperado em função que a ferramenta não possui recursos para apoiar o processo de gerência de requisitos.

6 MÓDULO DE GERÊNCIA DE REQUISITOS

6.1 INTRODUÇÃO

PLACES é uma plataforma colaborativa desenvolvida para apoiar a gerência de projetos de software, flexibilizando a construção de ferramentas que venham oferecer suportar diferentes processos da Engenharia de Software Moretto (2005). Esta plataforma é uma evolução da ferramenta colaborativa ARCase, desenvolvida inicialmente para suportar apenas a fase de concepção de software (KONOWALOW, 2003; DOMINGOS, 2004). A proposta foi desenvolver um ambiente comum para a construção incremental de diferentes ferramentas onde os recursos colaborativos já estão disponíveis para esta plataforma. Desta forma, a construção das ferramentas foi facilitada e focada nas boas práticas, associadas com os processos que elas pretendem trabalhar. No trabalho de Moretto (2005), foi desenvolvida uma ferramenta para gerência de projetos, incluindo as áreas de planejamento e monitoração de projetos. No trabalho de Souza (2005), foi desenvolvida uma ferramenta para estimativa de projetos de software baseada na técnica de pontos por caso de uso. Da mesma forma, foi desenvolvida toda uma infra-estrutura de segurança e de colaboração (ferramentas para reuniões virtuais, fóruns de discussão e outros recursos). A figura 4 apresenta uma tela da plataforma PLACES, onde é possível visualizar os projetos em andamento.

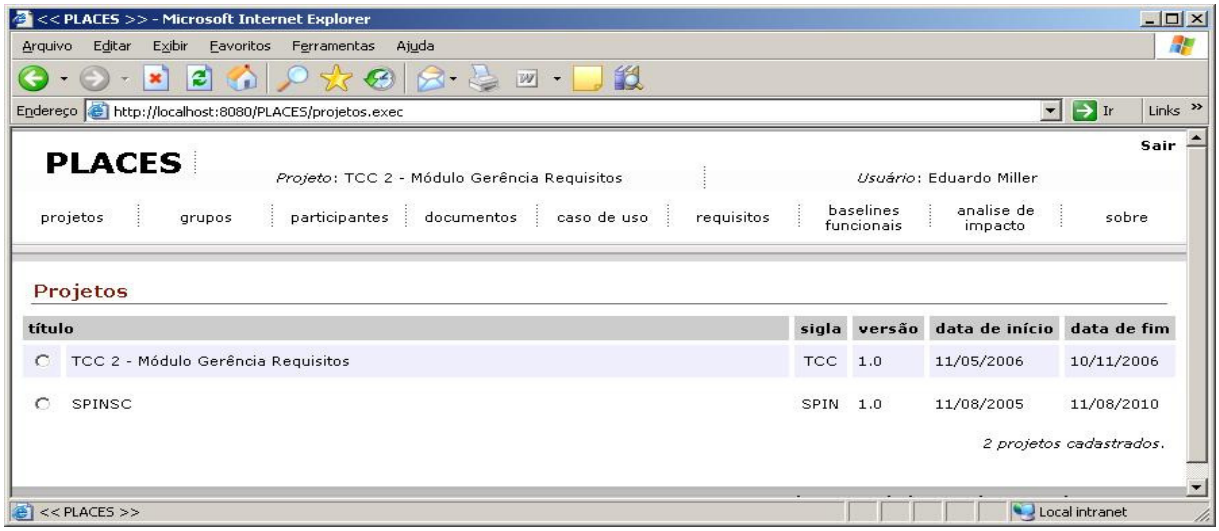

<< PLACES >> - Microsoft Internet Explorer
Arquivo Editar Exibir Favoritos Ferramentas Ajuda
Endereço http://localhost:8080/PLACES/projetos.exec
Ir Links
PLACES
Sair
Projeto: TCC 2 - Módulo Gerência Requisitos
Usuário: Eduardo Miller
projetos grupos participantes documentos caso de uso requisitos baselines funcionais analise de impacto sobre
Projetos
título sigla versão data de início data de fim
TCC 2 - Módulo Gerência Requisitos TCC 1.0 11/05/2006 10/11/2006
SPINSC SPIN 1.0 11/08/2005 11/08/2010
2 projetos cadastrados.
<< PLACES >>
Local intranet

Figura 50 - Tela de projetos do PLACES.

Neste sentido, o ambiente ARCase passou a ser uma das ferramentas da plataforma PLACES, dando ênfase à área de desenvolvimento de requisitos. Um dos objetivos deste trabalho é a migração de parte (aquela associada à gerência de requisitos) da ferramenta ARCase na plataforma PLACES. A filosofia inicial do ambiente ARCase foi preservada nesta nova versão, onde a ênfase não está no desenvolvimento de diagramas, uma vez que ela não possui recursos para a construção de diagramas. O foco continua sendo a manipulação eficiente de documentos, usualmente deixada em segundo plano por ferramentas gráficas. Utilizando os recursos disponibilizados pela plataforma PLACES, a ferramenta atua como um banco de informações compartilhadas entre os participantes do projeto, por meio de um ambiente colaborativo. Utilizando-se desta colaboração, são gerados diversos tipos de artefatos como regras de negócio, requisitos funcionais, requisitos não funcionais, telas do sistema e outros ainda customizáveis. Do ponto de vista deste trabalho, o requisito é o artefato central da ferramenta porém, o *framework* do PLACES permite associar quaisquer artefatos construídos entre si e, com os requisitos, oferecendo rastreabilidade.

Entretanto, a versão anterior da ferramenta não possuía nenhum recurso para auxiliar os gerentes na gerência dos requisitos do projeto. Sendo assim, este trabalho buscou um objetivo (motivação principal para este TCC) que foi o desenvolvimento de um módulo de gerência de requisitos de software com base nos artefatos disponibilizados pela ferramenta de desenvolvimento de requisitos da plataforma PLACES. A partir deste módulo, é possível gerenciar os requisitos do projeto. A atividade de controle de alterações em requisitos, uma das etapas do processo de gerência de requisitos não será abordada por este trabalho porque já é tema de outro TCC de José Paulo, deste curso.

6.2 O MÓDULO DE GERÊNCIA DE REQUISITOS

O módulo desenvolvido dá suporte às atividades de cadastramento dos requisitos do projeto, manutenção das *baselines* de requisitos e disponibiliza uma matriz de rastreabilidade entre requisitos para auxílio na análise se impacto do projeto. As próximas seções descrevem como as tarefas mencionadas anteriormente podem ser executadas no sistema.

6.2.1 Cadastro de requisitos

Para que a Gerência de Requisitos possa ser demonstrada pela ferramenta, foi necessário disponibilizar um recurso para a definição dos requisitos. Embora a versão atual permita a edição de casos de uso, este trabalho buscou deixar a ferramenta mais flexível com a possibilidade de cadastrar requisitos de forma livre. A gerência considera que os requisitos já estão definidos. Para isso, a nova versão do sistema disponibiliza um cadastro para os requisitos do projeto.

6.2.1.1 Funcionamento

Na página que lista os requisitos do sistema (Figura 51) é possível visualizar todos os requisitos do projeto. Desta página é possível acessar as páginas de inclusão e edição, além de poder excluir um requisito.

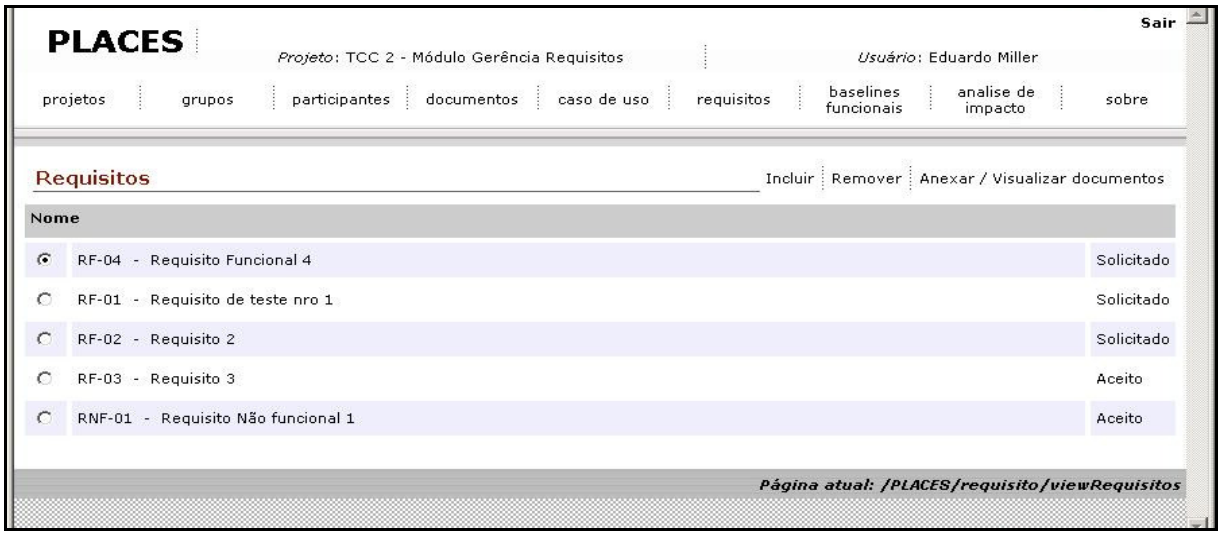


Figura 51 - Listagem dos requisitos cadastrados.

Na página que lista os requisitos do sistema, ao optar pela opção "*Incluir*" o sistema exibirá a página de cadastro do requisito (Figura 52).

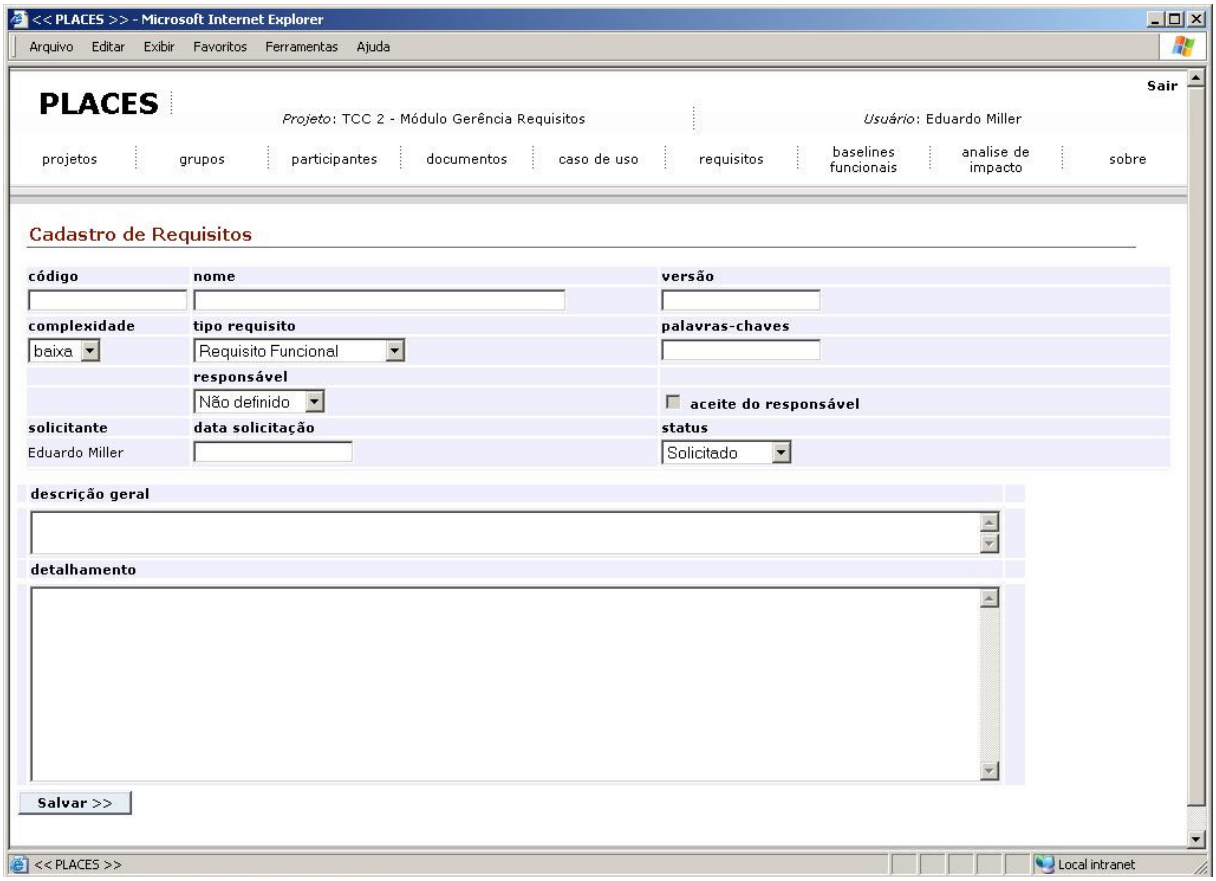


Figura 52 - Tela de cadastro de um novo requisito.

Na página de cadastro do requisito, no campo "*código*" o usuário pode escrever um código para o requisito. No campo "*nome*" o usuário pode escrever o nome do requisito a ser cadastrado. No campo "*versão*" o usuário pode escrever a versão do requisito (o valor padrão é 1.0). No campo "*complexidade*" o usuário pode selecionar a complexidade do ator a ser cadastrado (que poderá será usado para o cálculo de estimativas de custo e pode variar entre "baixo", "médio" e "alto"). No campo "*tipo requisito*" o usuário pode selecionar outro tipo de requisito para ser cadastrado (o valor padrão é "Requisito Funcional", mas o usuário pode optar por cadastrar "Requisitos Não Funcionais"). No campo "*status*", o usuário pode o usuário pode selecionar o status do requisito para ser cadastrado (o valor padrão para um novo requisito é "Solicitado"). No campo "*palavras-chave*", o usuário pode escrever as palavras-chave que representam o requisito a ser cadastrado. No campo "*responsável*", o usuário pode escolher o participante o qual será considerado o responsável pelo requisito a ser cadastrado. No campo "*autor*", o usuário pode ver o nome do usuário logado no sistema. O campo "*aceite responsável*" tem como finalidade registrar o aceite da atribuição de responsabilidade do requisito e é desabilitado no cadastramento de um novo requisito. No campo "*descrição geral*" o usuário pode escrever uma breve descrição do requisito a ser cadastrado. O botão

"*Salvar*" serve para salvar o novo requisito no sistema. Ao escolher a opção "*Salvar*" o sistema salva o requisito, exibindo novamente a tela da listagem dos requisitos.

Ao clicar sobre um requisito, na página que lista os requisitos, o sistema exibe a página de cadastro do requisito, com os campos preenchidos de acordo com o que está cadastrado no banco de dados. Se o usuário alterar qualquer um dos campos e selecionar a opção "*Salvar*", as informações serão atualizadas no banco de dados.

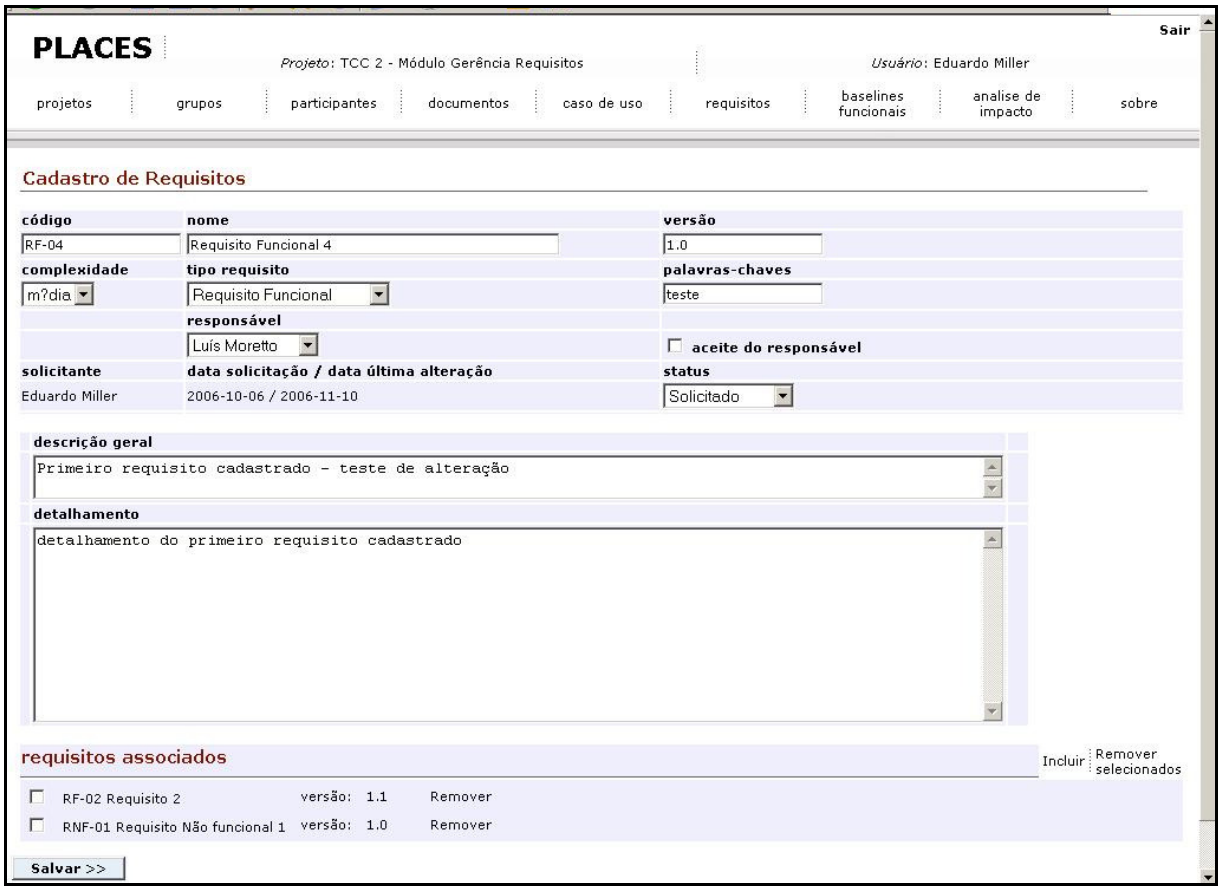


Figura 53 - Detalhes de um requisito cadastrado.

O campo "*aceite responsável*" tem como finalidade registrar o aceite da atribuição de responsabilidade do requisito. Este campo é desabilitado no cadastramento de um novo requisito.

Na Figura 53, existe uma região onde são listados os requisitos que estão associados ao requisito que está sendo exibido. Esta região é exibida somente na tela de detalhes do requisito. À direita estão as opções para incluir associações entre o requisito exibido e os demais requisitos ("*Incluir*") e ou remover alguma associação entre o requisito exibido e seus

requisitos associados. Este recurso tem como finalidade estabelecer uma rastreabilidade entre os requisitos.

Para incluir uma associação entre requisitos, escolhe-se a opção "*Incluir*" que exibirá a tela de requisitos a serem associados ao requisito que está sendo exibido (Figura 54).

Figura 54 - Listagem de requisitos para associação.

Para realizar a associação dos requisitos, basta selecionar os requisitos a serem associados e clicar o botão "Associar Requisitos". Após esta operação, o sistema criará uma associação entre o requisito que está identificado na parte superior da tela e os requisitos selecionados. Em seguida, o usuário é direcionado à tela de detalhamento do requisito. Esta associação entre requisitos permite que tenhamos uma rastreabilidade horizontal entre os requisitos do sistema.

A exclusão das associações criadas pelo processo anterior é feita na tela de detalhamento do requisito selecionando-se os requisitos associados e clicando sobre "Remover Selecionados". A partir desta operação, o sistema irá excluir a associação entre o requisito que está sendo exibido e os requisitos associados selecionados. A Figura 55 destaca esta área da tela de detalhamento do requisito.

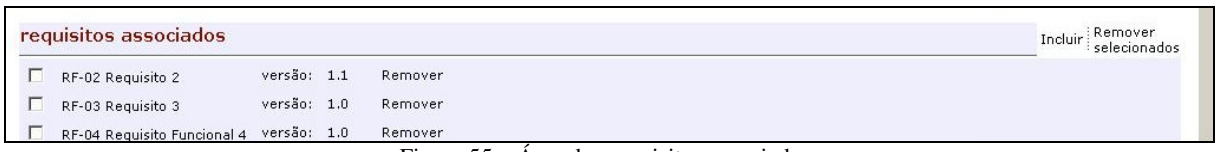

requisitos associados
Incluir
Remover selecionados
RF-02 Requisito 2 versão: 1.1 Remover
RF-03 Requisito 3 versão: 1.0 Remover
RF-04 Requisito Funcional 4 versão: 1.0 Remover

Figura 55 – Área dos requisitos associados.

Um requisito X que foi associado durante o cadastro do requisito Y, aparecerá como requisito associado na tela de detalhes do requisito Y. Quando for aberta a tela com os detalhes do requisito X, o requisito Y também aparecerá como associado para demonstrar a bidirecionalidade da associação. Esta ligação, sendo bidirecional, tem um comportamento similar na exclusão da associação.

Ao selecionar um requisito na página que lista os requisitos e logo após optar pela opção "*Excluir*", (Figura 56) o sistema exibe uma tela com a pergunta "Você deseja excluir o requisito selecionado?". Se clicar o botão "*OK*" o sistema irá apagar o requisito da base de dados e voltar para a página que lista os requisitos (Figura 51). Se for pressionado o botão "*Cancelar*" o sistema irá retornar a página que lista os requisitos sem alterar a base de dados.

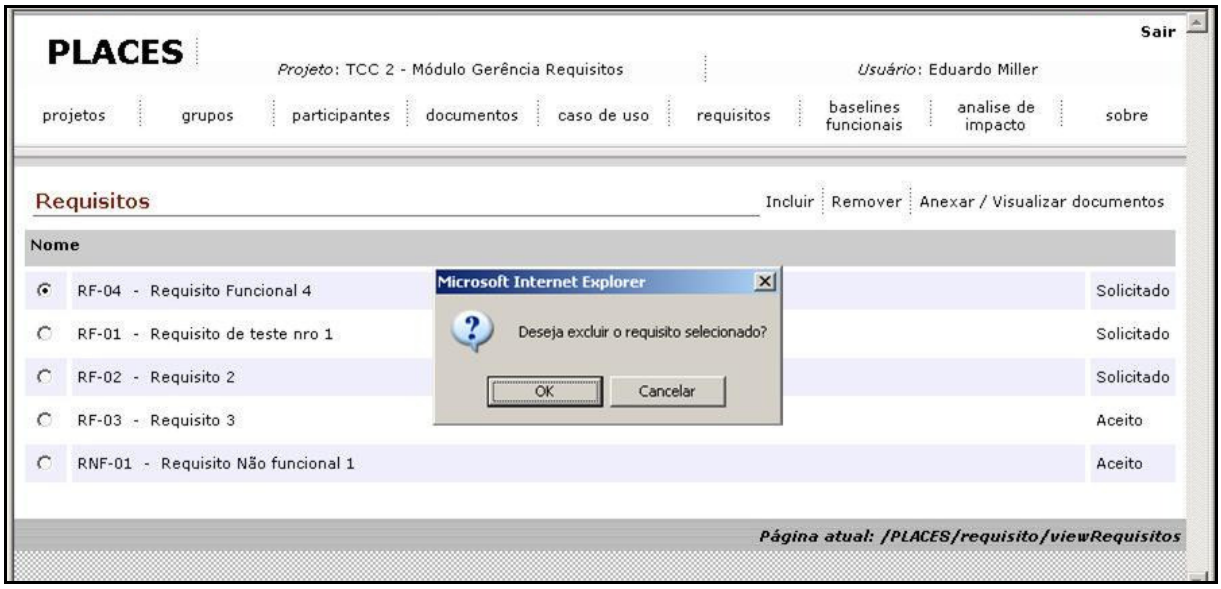


Figura 56 - Exclusão de requisito.

6.2.1.2 Documentação do Cadastro de Requisitos

Nesta seção são detalhados os casos de uso do cadastro de requisitos do PLACES. O cadastro de requisitos realiza dois casos de uso: o caso de uso de cadastro de requisitos e o caso de uso de associação de requisitos.

Caso de Uso 001 - Nome: **Cadastro de Requisitos**	
Ator ativo: Analista de Sistema	**Atores passivos**: N\A.
Objetivo: Cadastrar requisitos no sistema	
Pré-condição: O analista de sistema deve estar logado no sistema.	
FLUXO BASE 1. O sistema exibe a lista de requisitos. (Figura 51). 2. O analista seleciona a opção incluir. 3. O sistema apresenta a tela de cadastro de requisitos (Figura 52). 4. O analista digita os dados do requisito, e opta por gravá-los. 5. O sistema grava os dados do requisito e mostra a lista de requisitos (Figura 51).	

Pós-condição: O requisito está cadastrado no sistema.
FLUXO ALTERNATIVO – FA-01 Cancelamento 1. O sistema sai da tela de cadastro de requisitos. **Pós-condição**: Nenhuma alteração é feita no sistema.
FLUXO ALTERNATIVO – FA-2 Edita requisito No passo 2 do fluxo base, o analista clica sobre um requisito. 2. O sistema exibe uma tela de edição de requisito (Figura 53). 3. O analista edita o requisito e opta por gravá-lo. 4. O sistema grava os dados do requisito e mostra a lista de requisitos (Figura 51). **Pós-Condição**: O requisito foi alterado.
FLUXO ALTERNATIVO – FA-3 Exclui requisito No passo 2 do fluxo base, o analista seleciona um requisito. 1. O analista seleciona a opção Excluir. 2. O sistema exibe uma tela de confirmação (Figura 56). 3. O analista confirma. 4. O sistema exclui o requisito. 5. O sistema exibe uma mensagem de exclusão feita com sucesso. **Pós-Condição:** O requisito foi excluído**.**
FLUXO DE EXCEÇÃO – FE-1 Erro ao salvar requisito No passo 4 do fluxo base, o sistema não consegue salvar o requisito. 1. O sistema sai da tela de cadastro de requisito e apresenta uma tela de erro. **Pós-condição**: Nenhuma alteração no sistema
FLUXO DE EXCEÇÃO – FE-2 Erro ao editar requisito No passo 3 do fluxo alternativo 2, o sistema não consegue salvar as alterações . 1. O sistema exibe uma tela de erro.
FLUXO DE EXCEÇÃO – FE-3 Erro ao excluir No passo 5 do fluxo alternativo 3, o sistema não consegue excluir o requisito. 1. O sistema exibe uma tela de erro. **Pós-Condição:** Nenhuma alteração no sistema.

Tabela 13 - Caso de Uso Cadastro de Requisitos.

A seguir é detalhado o caso de uso de associação de requisitos.

Caso de Uso 001 - Nome: **Associação de requisitos**	
Ator ativo: Analista de Sistema	**Atores passivos**: N\A.
Objetivo: Criar associação entre requisitos para manter rastreabilidade.	
Pré-condição: O analista de sistema deve estar logado no sistema.	
FLUXO BASE 1. O sistema exibe a lista de requisitos. (Figura 51). 2. O analista clica sobre o requisito que deseja associar outros requisitos. 3. O sistema apresenta a tela de cadastro de requisitos (Figura 52). 4. O analista escolhe a opção “Incluir” na área de associação de requisitos. 5. A lista de requisitos para associação é mostrada; 6. O analista escolhe os requisitos que quer associar e escolhe “Associar requisitos”. 5. O sistema grava os dados da associação do requisito e mostra a tela de cadastro de requisitos (Figura 52). **Pós-condição**: A associação entre os requisitos está cadastrada no sistema.	
FLUXO ALTERNATIVO – FA-1 Exclui associação requisito No passo 4 do fluxo base, o analista seleciona um requisito associado. 1. O analista seleciona a opção “Remover selecionados” na área de associação de requisitos. 2. O sistema exclui a associação do requisito e mostra a tela de cadastro de requisitos (Figura 52). **Pós-Condição:** A associação entre os requisitos foi excluída do sistema**.**	
FLUXO DE EXCEÇÃO – FE-1 Erro ao salvar associação entre os requisitos No passo 5 do fluxo base, o sistema não consegue salvar a associação entre os requisitos. 1. O sistema sai da tela de cadastro de requisito e apresenta uma tela de erro. **Pós-condição**: Nenhuma alteração no sistema	
FLUXO DE EXCEÇÃO – FE-2 Erro ao excluir associação entre os requisitos No passo 2 do fluxo alternativo 1, o sistema não consegue excluir a associação entre os requisito.	

1. O sistema exibe uma tela de erro. **Pós-Condição:** Nenhuma alteração no sistema.

Tabela 14 - Caso de Uso Associação entre requisitos.

As figuras Figura 57 e Figura 58 mostram os diagramas de seqüência dos dois casos de uso do cadastro de requisitos.

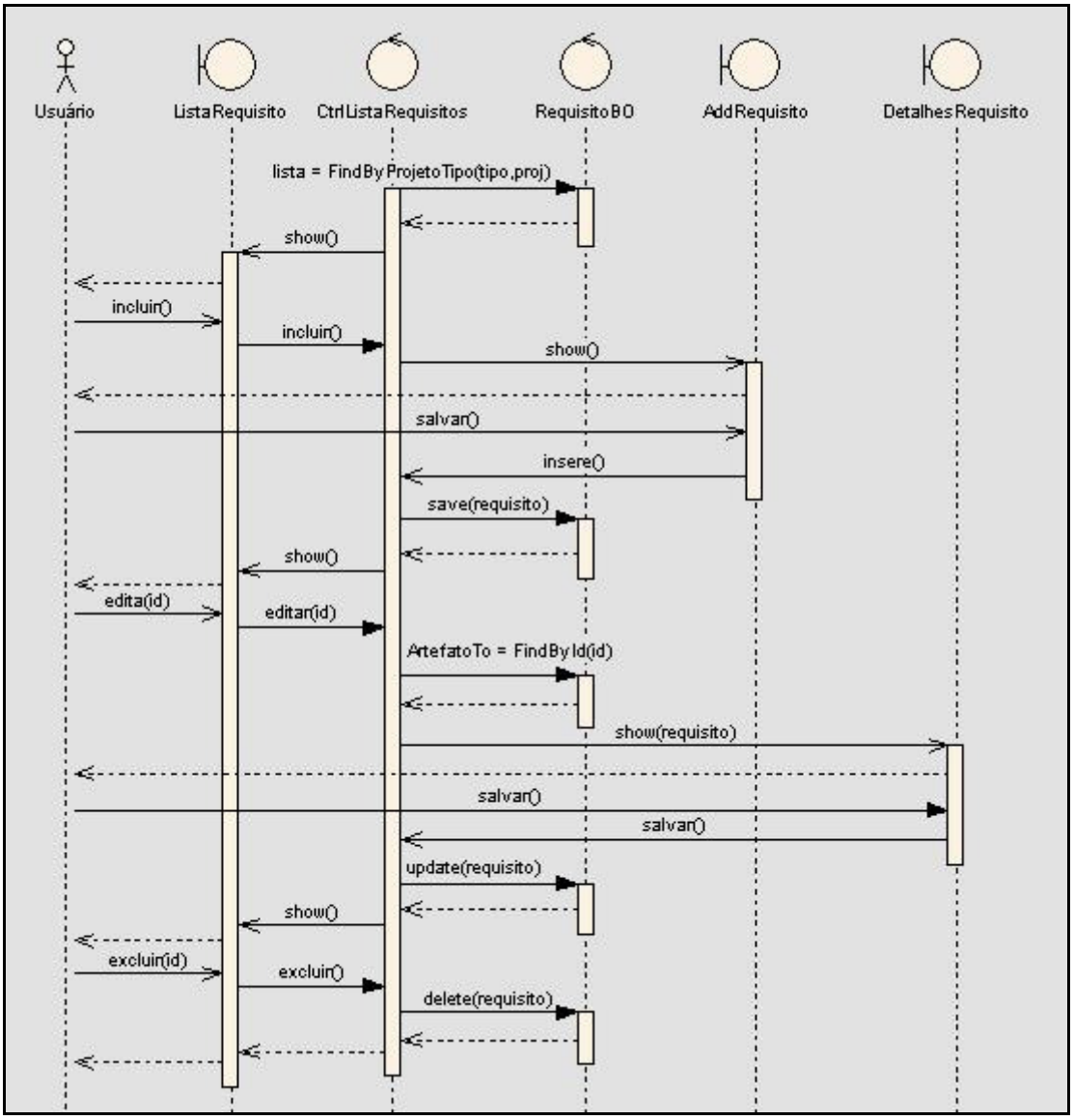


Figura 57 - Diagrama de Seqüência Cadastro de Requisitos

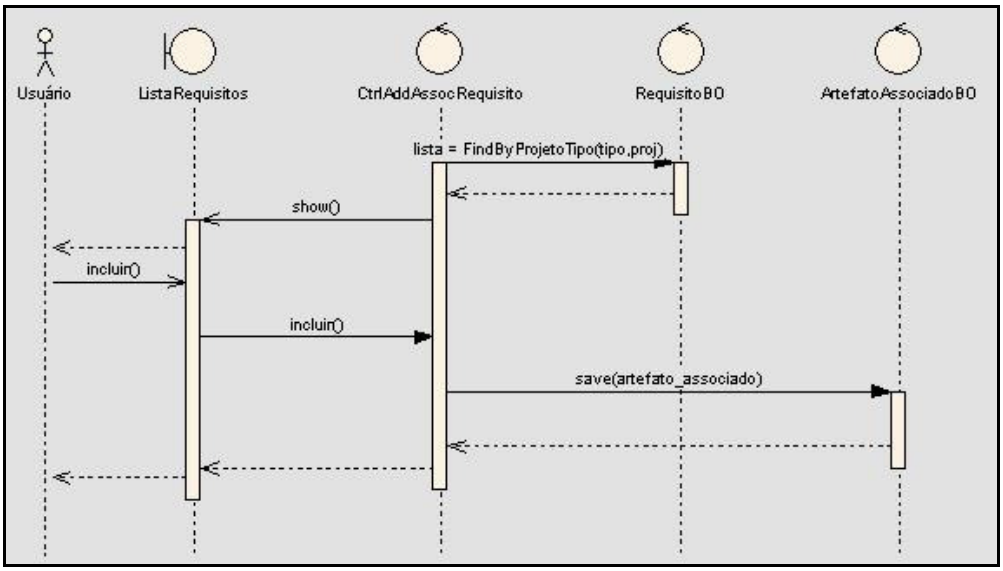


Figura 58 - Diagrama de Seqüência Associação entre Requisitos

6.2.2 Cadastro de Baselines

Uma das melhorias implementadas no PLACES foi o cadastramento de *baselines*. Com este recurso é possível criar uma *baseline* e definir quais requisitos fazem parte desta. Desta forma podemos ter uma rastreabilidade entre requisitos de forma a atender, parcialmente, uma das práticas específicas de Gerência de Requisitos do CMMI. Nesta versão do PLACES, é possível colocar apenas requisitos em uma *baseline*.

6.2.2.1 Funcionamento

Na página que lista as *baselines* do sistema (Figura 59) é possível visualizar todas as *baselines* do projeto. Desta página é possível acessar as páginas de inclusão e edição, além de poder excluir uma *baseline*.

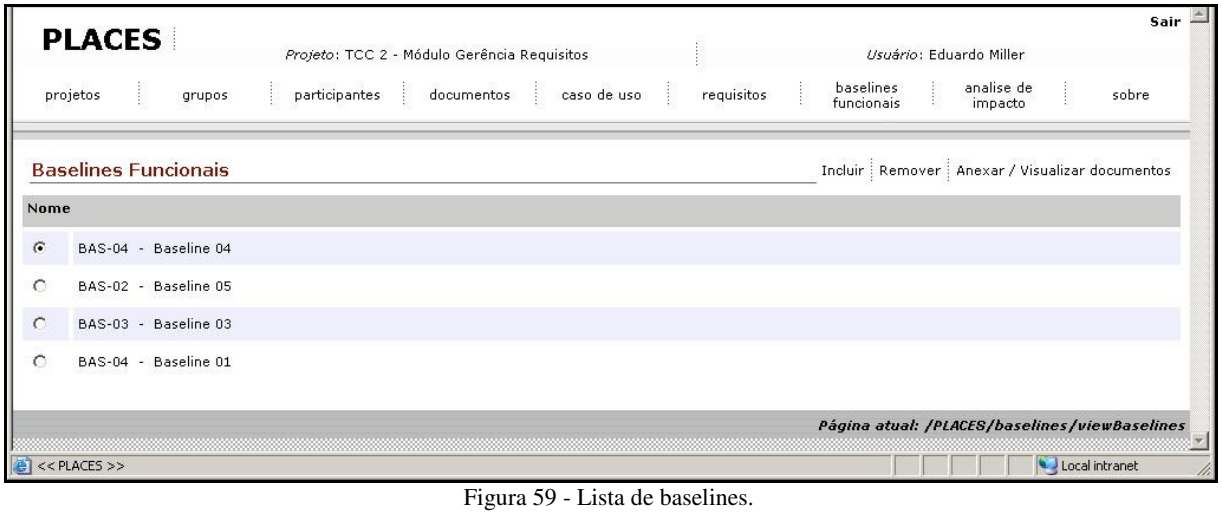


Figura 59 - Lista de baselines.

Na página que lista as baselines do sistema, ao optar pela opção "*Incluir*" o sistema exibirá a página de cadastro do requisito (Figura 60).

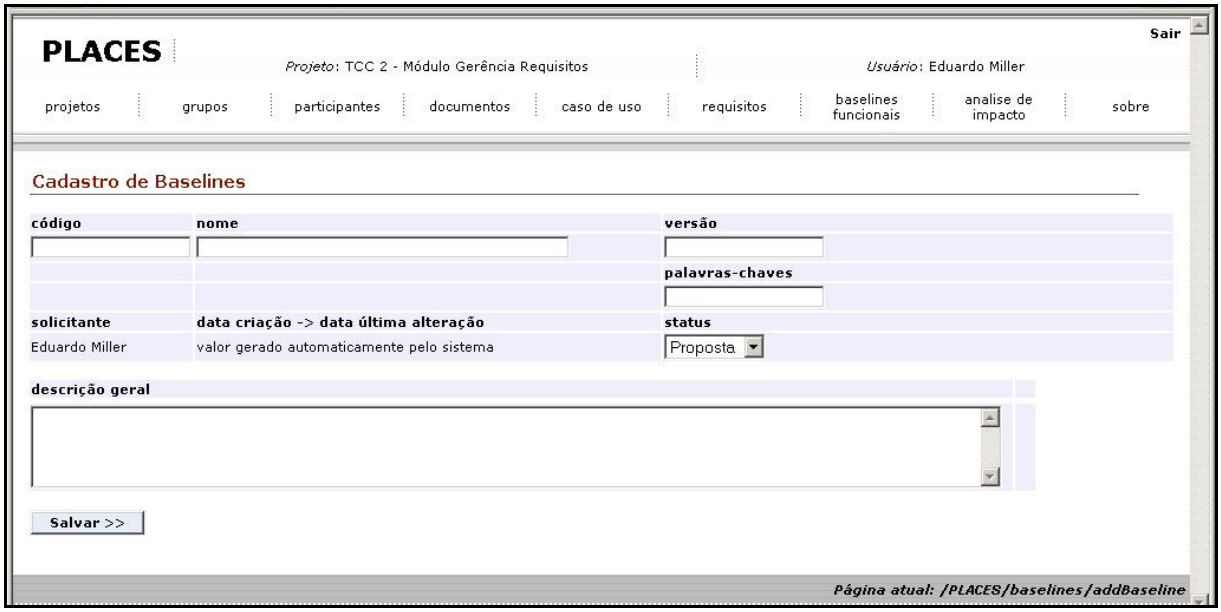


Figura 60 - Tela de cadastro de uma nova *baseline.*

Na página de cadastro da *baseline*, no campo "*código*" o usuário pode escrever um código para a *baseline*. No campo "*nome*" o usuário pode escrever o nome da *baseline* a ser cadastrada. No campo "*versão*" o usuário pode escrever a versão da *baseline* (o valor padrão é 1.0). No campo "*palavras-chave*", o usuário pode escrever as palavras-chave que representam a *baseline* a ser cadastrada. O campo "*solicitante*" é automaticamente preenchido com o nome do usuário logado. Os campos "*data criação*" e "*data última alteração*" têm seus valores gerados automaticamente. No campo "*status*", o usuário pode o usuário pode selecionar o status da *baseline* para ser cadastrado (o valor padrão para uma nova *baseline* é "Proposta"). No campo "*descrição geral*" o usuário pode escrever uma breve descrição da *baseline* a ser cadastrada. O botão "*Salvar*" serve para salvar a nova *baseline* no sistema. Ao escolher a opção "*Salvar*" o sistema salva a *baseline*, exibindo novamente a tela da listagem das *baselines*.

Ao clicar sobre uma *baseline*, na página que lista as *baselines*, o sistema exibe a página de cadastro da *baseline*, com os campos preenchidos de acordo com o que está cadastrado no banco de dados (Figura 61). Se o usuário alterar qualquer um dos campos e selecionar a opção "*Salvar*", as informações serão atualizadas no banco de dados.

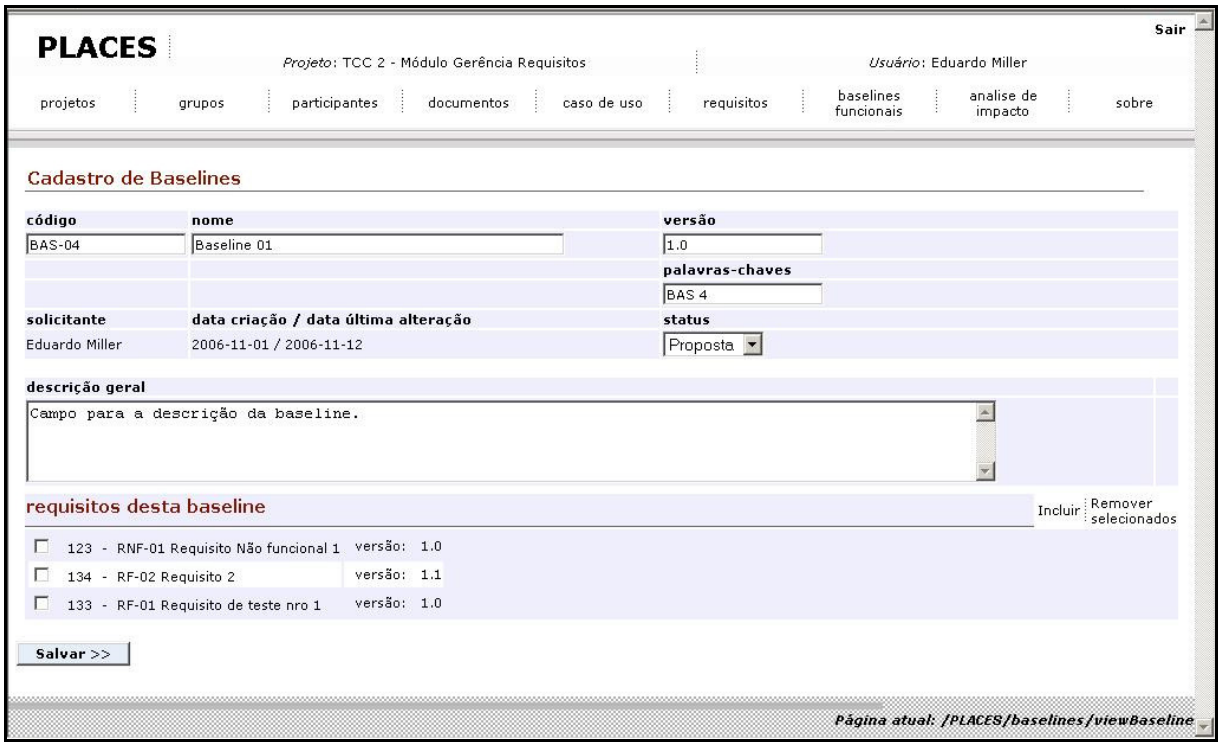


Figura 61 - Detalhes de uma baseline.

Para a inclusão de requisitos em uma *baseline*, é necessário exibir os detalhes da *baseline*. A Figura 62, existe a região onde são listados os requisitos que fazem parte da *baseline* que está sendo exibida. Esta região é exibida somente na tela de detalhes da *baseline*. À direita estão as opções para incluir requisitos na *baseline* ("*Incluir*") e ou remover algum requisito da *baseline*. Este recurso tem como finalidade estabelecer *baselines* de requisitos do sistema, com o objetivo de atender alguns aspectos da prática genérica 2.6 da PA Gerência de Requisitos do CMMI.

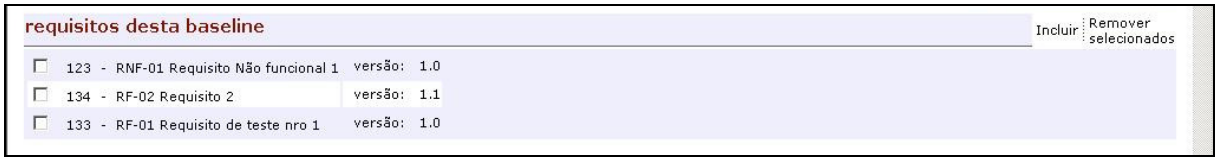

requisitos desta baseline
Incluir
Remover selecionados
123 - RNF-01 Requisito Não funcional 1 versão: 1.0
134 - RF-02 Requisito 2 versão: 1.1
133 - RF-01 Requisito de teste nro 1 versão: 1.0

Figura 62 - Requisitos da *baseline.*

Para incluir um requisito na *baseline*, escolhe-se a opção "*Incluir*" que exibirá a tela de requisitos para inclusão na *baseline* (Figura 63).

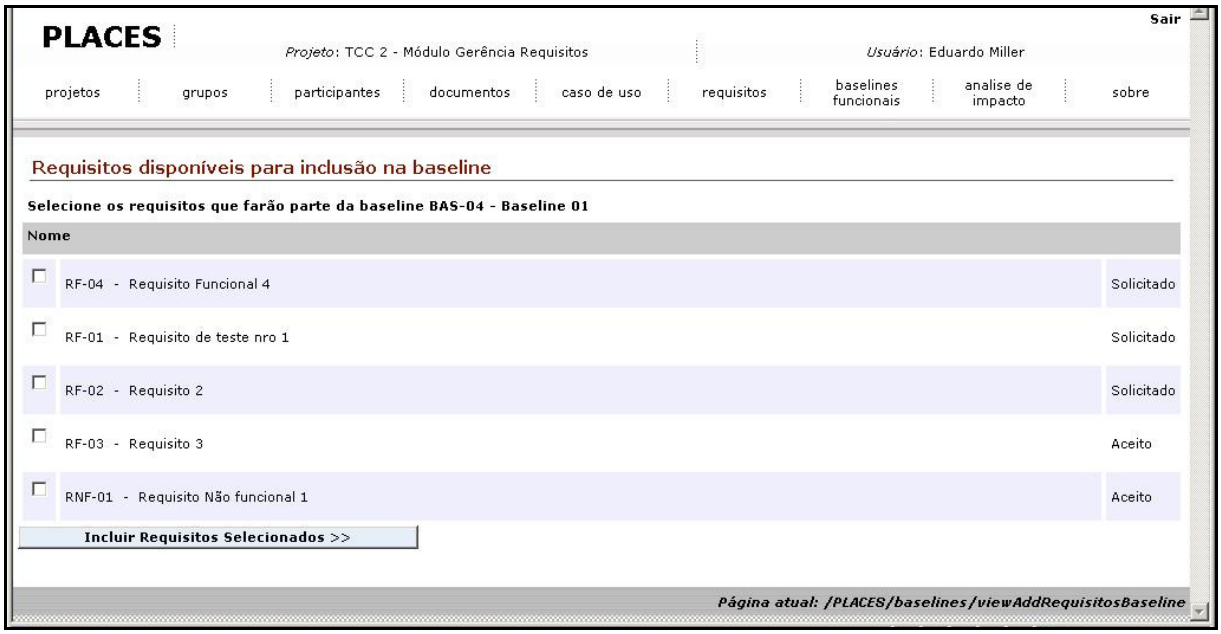


Figura 63 - Requisitos para inclusão na *baseline.*

Para um requisito na *baseline*, basta selecionar os requisitos e clicar o botão "Incluir Requisitos Selecionados". Após esta operação, o sistema incluirá os requisitos na baseline está identificada na parte superior da tela. Em seguida, o usuário é direcionado à tela de detalhamento da *baseline*.

A exclusão dos requisitos incluídos pelo processo anterior é feita na tela de detalhamento da *baseline* selecionando-se os requisitos a serem excluídos e clicando sobre "Remover Selecionados". A partir desta operação, o sistema irá excluir os requisitos selecionados da *baseline* que está sendo exibida. A Figura 62 destaca esta área da tela de detalhamento da *baseline*.

Ao selecionar uma *baseline* na página que lista as *baselines* e logo após, o usuário, optar pela opção "*Excluir*", (Figura 64) o sistema exibe uma tela com a pergunta "Você deseja excluir a *baseline* selecionada?". Se pressionar o botão "*OK*" o sistema irá apagar a baseline da base de dados e voltar para a página que lista as baselines (Figura 59). Se pressionar o botão "*Cancelar*" o sistema irá retornar a página que lista as *baselines* sem alterar a base de dados.

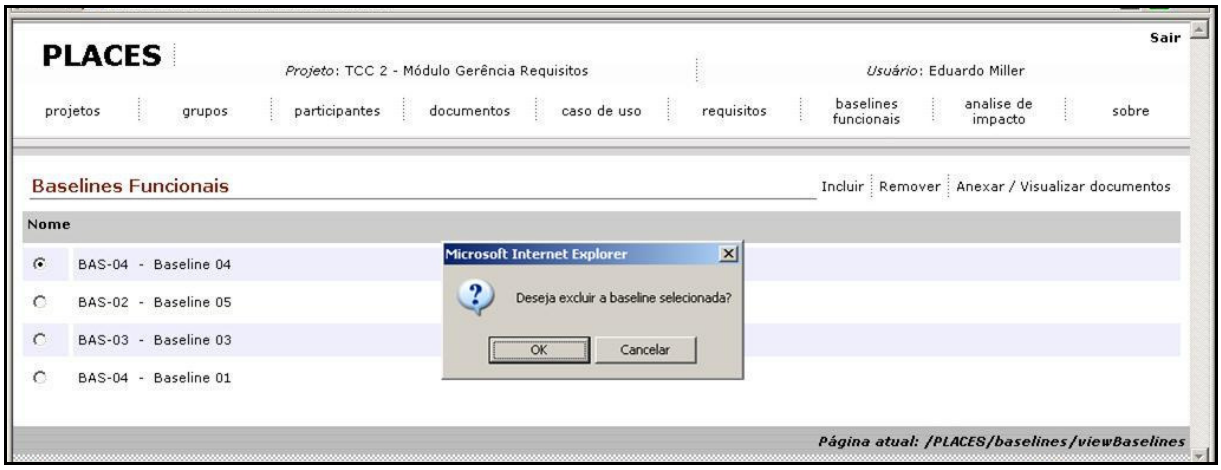


Figura 64 - Exclusão de baselines.

6.2.2.2 Documentação do Cadastro de Baselines

Nesta seção são detalhados os casos de uso do cadastro de *baselines* do PLACES. O cadastro de *baselines* realiza dois casos de uso: o caso de uso de cadastro de *baselines* e o caso de uso de inclusão de requisitos de *baselines*.

Caso de Uso 001 - Nome: **Cadastro de Baselines**	
Ator ativo: Analista de Sistema	**Atores passivos**: N\A.
Objetivo: Cadastrar baselines no sistema	
Pré-condição: O analista de sistema deve estar logado no sistema.	
FLUXO BASE 1. O sistema exibe a lista de baselines. (Figura 59). 2. O analista seleciona a opção incluir. 3. O sistema apresenta a tela de cadastro de baselines (Figura 60). 4. O analista digita os dados da baseline, e opta por gravá-los. 5. O sistema grava os dados da baseline e mostra a lista de baselines (Figura 59). **Pós-condição**: A baseline está cadastrada no sistema.	
FLUXO ALTERNATIVO – FA-01 Cancelamento 1. O sistema sai da tela de cadastro de baselines. **Pós-condição**: Nenhuma alteração é feita no sistema.	
FLUXO ALTERNATIVO – FA-2 Edita baseline No passo 2 do fluxo base, o analista clica sobre uma baseline. 2. O sistema exibe uma tela de edição de baseline (Figura 61). 3. O analista edita a baseline e opta por gravá-lo. 4. O sistema grava os dados da baseline e mostra a lista de baselines (Figura 59). **Pós-Condição**: O requisito foi alterado.	
FLUXO ALTERNATIVO – FA-3 Exclui baseline No passo 2 do fluxo base, o analista seleciona uma baseline. 1. O analista seleciona a opção Excluir. 2. O sistema exibe uma tela de confirmação. 3. O analista confirma. 4. O sistema exclui a baseline. 5. O sistema exibe uma mensagem de exclusão feita com sucesso. **Pós-Condição:** A baseline foi excluída**.**	
FLUXO DE EXCEÇÃO – FE-1 Erro ao salvar baseline No passo 4 do fluxo base, o sistema não consegue salvar a *baseline*. 1. O sistema sai da tela de cadastro da baseline e apresenta uma tela de erro. **Pós-condição**: Nenhuma alteração no sistema	
FLUXO DE EXCEÇÃO – FE-2 Erro ao editar baseline	

No passo 3 do fluxo alternativo 2, o sistema não consegue salvar as alterações . 1. O sistema exibe uma tela de erro.
FLUXO DE EXCEÇÃO – FE-3 Erro ao excluir No passo 5 do fluxo alternativo 3, o sistema não consegue excluir a baseline. 1. O sistema exibe uma tela de erro. **Pós-Condição:** Nenhuma alteração no sistema.

Tabela 15 - Caso de Uso Cadastro de Baselines.

A seguir é detalhado o caso de uso de inclusão de requisitos em uma *baseline*.

Caso de Uso 001 - Nome: **Inclusão de requisitos em uma baseline**	
Ator ativo: Analista de Sistema	**Atores passivos**: N\A.
Objetivo: Inclusão de requisitos em uma baseline.	
Pré-condição: O analista de sistema deve estar logado no sistema.	
FLUXO BASE 1. O sistema exibe a lista de baselines. (Figura 51). 2. O analista clica sobre a baseline que deseja incluir requisitos. 3. O sistema apresenta a tela de cadastro de requisitos (Figura 52). 4. O analista escolhe a opção "Incluir" na área de requisitos da baseline. 5. A lista de requisitos para inclusão na baseline é mostrada; 6. O analista escolhe os requisitos que quer incluir na baseline e escolhe "Incluir requisitos selecionados". 5. O sistema grava os dados da inclusão do requisito na baseline e mostra a tela de cadastro de baselines (Figura 51). **Pós-condição**: A inclusão dos requisitos na baseline está cadastrada no sistema.	
FLUXO ALTERNATIVO – FA-1 Exclui requisito da baseline No passo 4 do fluxo base, o analista seleciona um requisito da baseline. 1. O analista seleciona a opção "Remover selecionados" na área de requisitos da baseline. 2. O sistema exibe uma tela de confirmação (Figura 64). 3. O analista confirma. 4. O sistema exclui o requisito da baseline e mostra a tela de cadastro de baselines (Figura 52). **Pós-Condição:** A o requisito removido da baseline**.**	
FLUXO DE EXCEÇÃO – FE-1 Erro ao incluir requisitos em uma baseline No passo 5 do fluxo base, o sistema não consegue salvar a inclusão dos requisitos na baseline. 1. O sistema sai da tela de inclusão de requisitos na baseline e apresenta uma tela de erro. **Pós-condição**: Nenhuma alteração no sistema	
FLUXO DE EXCEÇÃO – FE-2 Erro ao excluir requisitos de uma baseline No passo 4 do fluxo alternativo 1, o sistema não consegue excluir os requisitos de uma baseline. 1. O sistema exibe uma tela de erro. **Pós-Condição:** Nenhuma alteração no sistema.	

Tabela 16 - Caso de Uso Inclusão de requisitos em uma *baseline.*

As figuras Figura 65 e Figura 66 mostram os diagramas de seqüência dos dois casos de uso do cadastro de *baselines*.

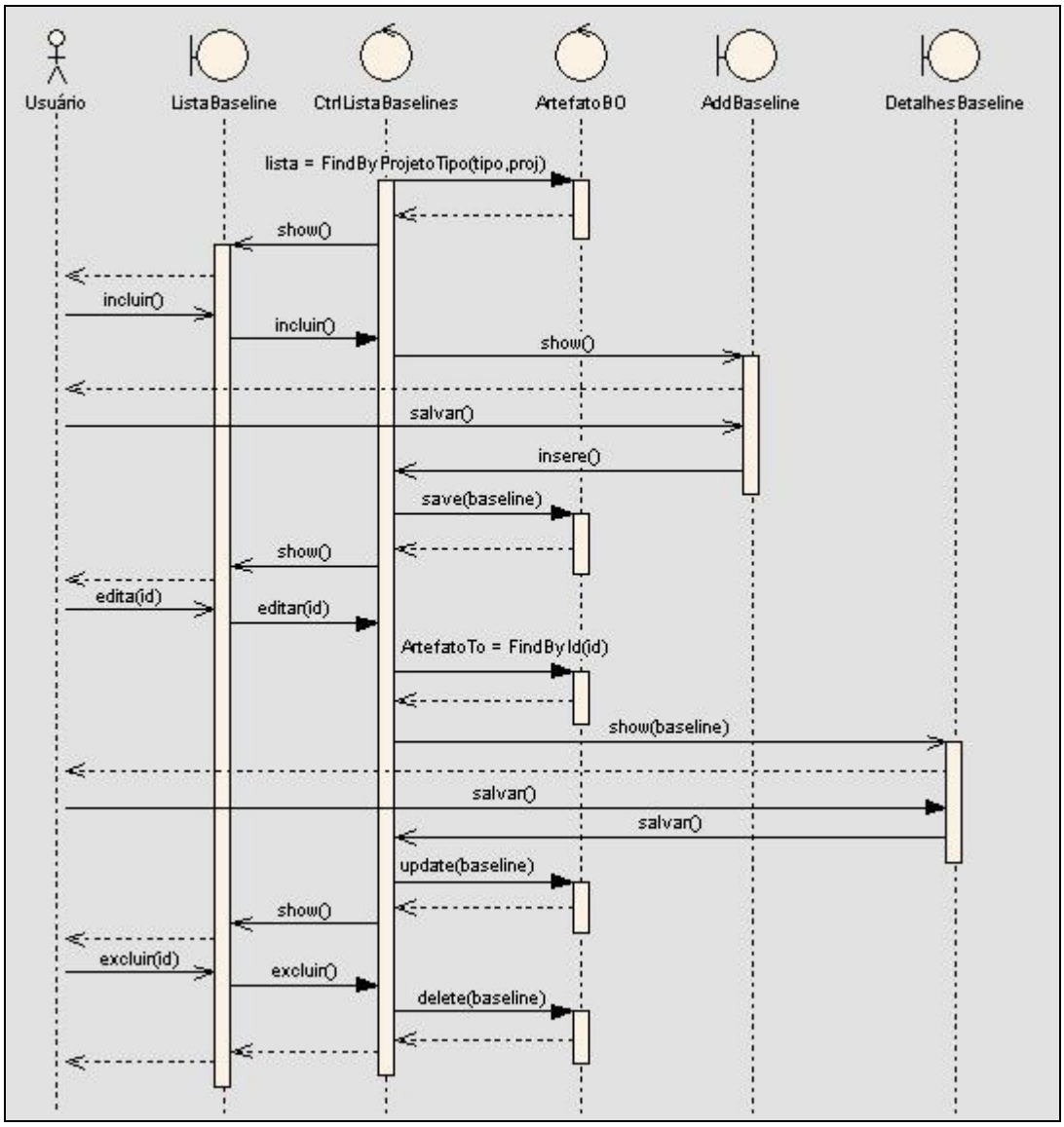


Figura 65 - Diagrama de Seqüência Cadastro de Baselines

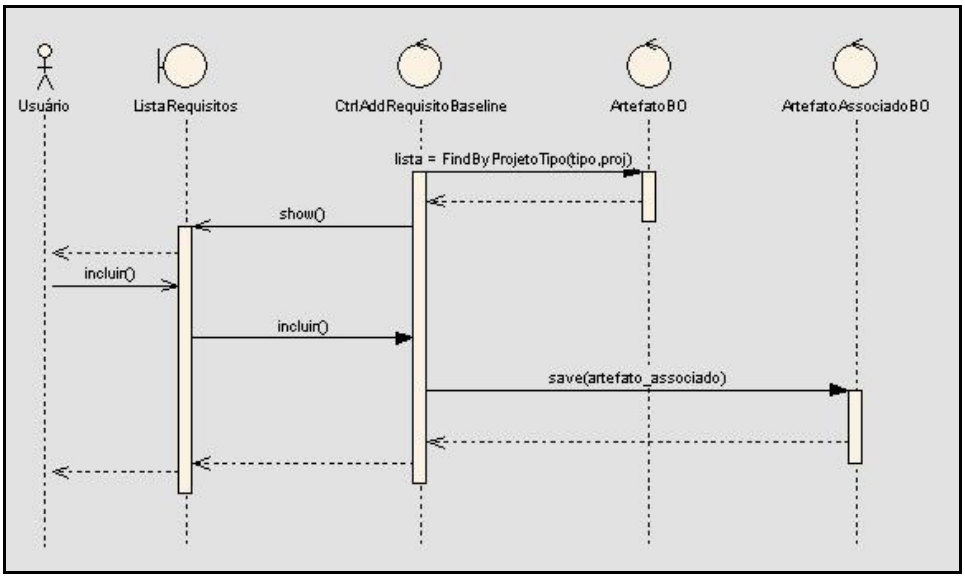


Figura 66 - Diagrama de Seqüência Inclusão de requisito em uma *baseline*

6.2.3 Matriz de Rastreabilidade

A matriz de rastreabilidade do PLACES foi criada com o objetivo de ser uma ferramenta para auxiliar na análise se impacto para o processo de controle de mudanças em requisitos.

Este trabalho não abordará o controle de mudanças em requisitos uma vez que este tema está sendo objetivo de outro Trabalho de Conclusão de Curso. Porém, como contribuição ao trabalho de controle de mudanças e para atender um dos objetivos deste trabalho, foi desenvolvida a matriz de rastreabilidade para visualizar as associações entre os requisitos.

A matriz desenvolvida neste trabalho, que mostra a rastreabilidade entre os requisitos do projeto, lista na horizontal e na vertical todos os requisitos cadastrados. Nas intersecções entre as linhas e colunas é indicada a associação entre os requisitos. Esta associação é indicada por uma seta verde (Figura 67). A matriz gerada pelo sistema mostras as associações entre os requisitos cadastradas pelo cadastro de requisitos (item 6.2.1).

6.2.3.1 Funcionamento

A matriz de rastreabilidade não possui parâmetro. Para acessar a matriz de rastreabilidade do PLACES o usuário deve escolher a opção “análise de impacto” no menu principal. A Figura 67 mostra a tela da matriz de rastreabilidade. A ligação entre os requisitos é mostrada, na intersecção das linhas e colunas, com uma seta verde. Desta forma é possível visualizar a rastreabilidade bidirecional entre os requisitos do projeto.

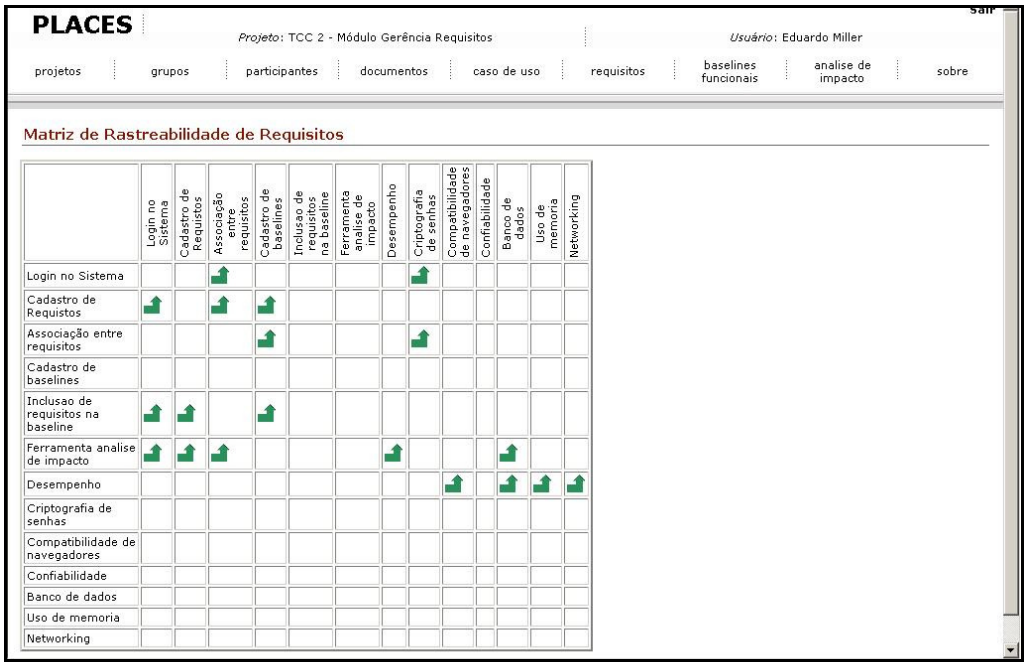


Figura 67 - Matriz de rastreabilidade do PLACES.

Na matriz, mostrada na Figura 67, pode-se observar, por exemplo, que o requisito “Inclusão de requisitos na baseline” está associado aos requisitos “Login no sistema”, “Cadastro de requisitos” e “Cadastro de baseline”. O exemplo citado pode ser mais bem visualizado na Figura 68. Este recurso permite observar a rastreabilidade entre os requisitos.

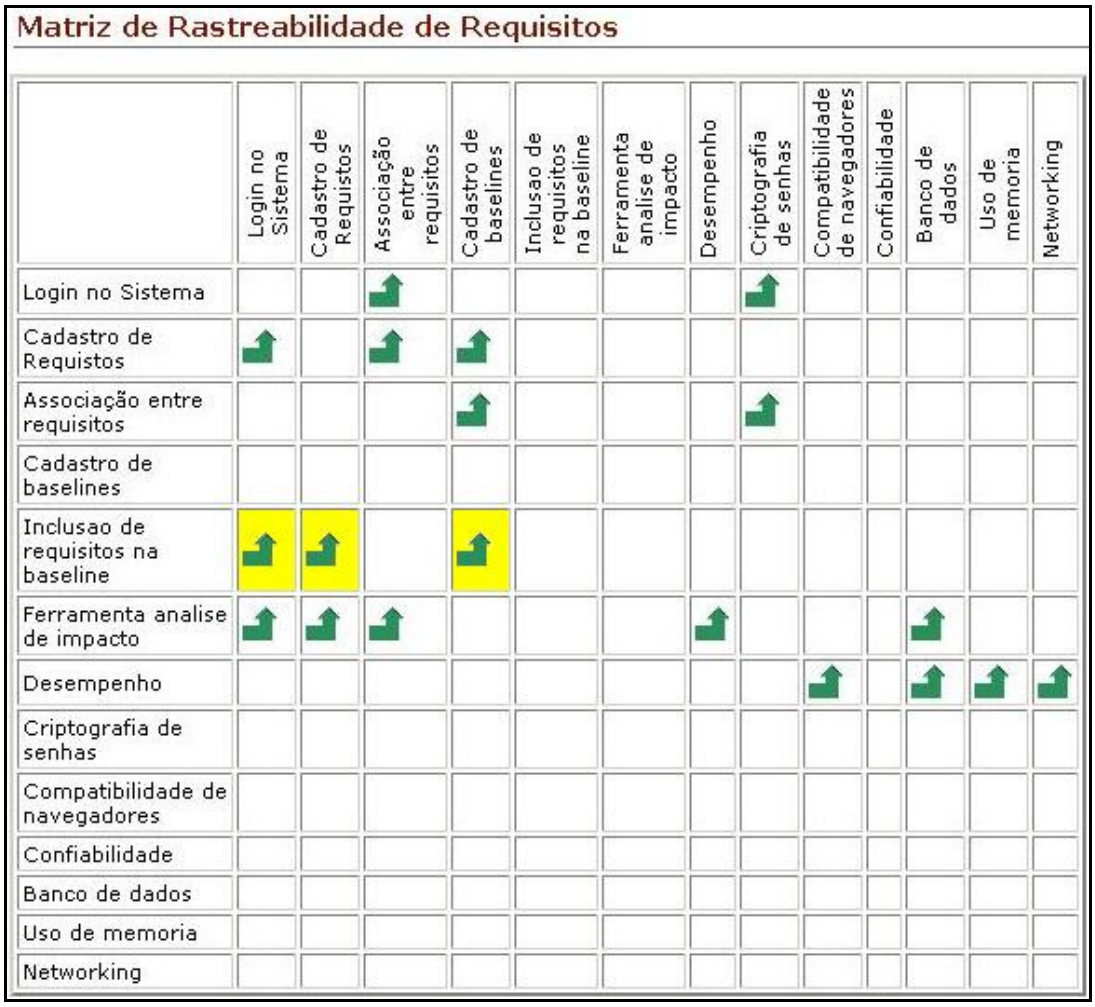


Figura 68 - Rastreabilidade vista na matriz de rastreabilidade.

6.2.3.2 Documentação da matriz de rastreabilidade

Nesta seção são detalhados o caso de uso da matriz de rastreabilidade do PLACES. A matriz de rastreabilidade um caso de uso: o caso de uso de visualização da matriz de rastreabilidade de requisitos do PLACES.

Caso de Uso 001 - Nome: **Visualização da Matriz de rastreabilidade**	
Ator ativo: Analista de Sistema	**Atores passivos**: N\A.
Objetivo: Gerar e mostrar a matriz de rastreabilidade de requisitos do sistema	
Pré-condição: O analista de sistema deve estar logado no sistema.	
FLUXO BASE 1. O analista seleciona a opção incluir. 2. O sistema gera a matriz de rastreabilidade e apresenta a tela com a matriz(Figura 67). **Pós-condição**: A baseline está gerada e mostrada pelo sistema.	
FLUXO DE EXCEÇÃO – FE-1 Erro ao gerar ou exibir a matriz de rastreabilidade No passo 2 do fluxo base, o sistema não consegue gerar ou exibir a matriz de rastreabilidade. 1. O sistema sai da tela da matriz de rastreabilidade e apresenta uma tela de erro. **Pós-condição**: Nenhuma alteração no sistema	

Tabela 17 - Caso de Uso Matriz de Rastreabilidade.

A Figura 69 mostra o diagrama de seqüência do caso de uso da geração e exibição da matriz de rastreabilidade do PLACES.

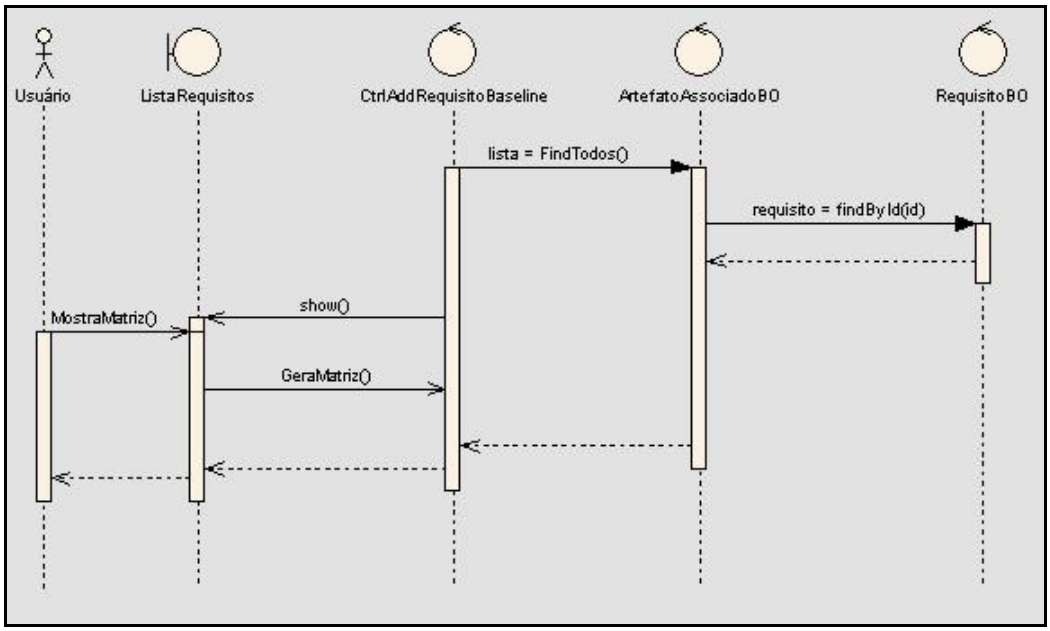


Figura 69 - Diagrama de Seqüência Matriz de Rastreabilidade

6.2.4 Anexar documentos

Como recurso adicional, foi reconstruída a opção de anexar arquivos externos ao projeto. Este recurso permite anexar ao projeto quaisquer documentos independentes do formato. Este recurso é importante para dar suporte a várias práticas genéricas do CMMI, uma ves que é possível anexar o Plano de Gerência de Requisitos, o documento de requisitos do sistema, relatórios de controle de mudanças e relatórios de status e avaliação dos requisitos do projeto.

6.2.4.1 Funcionamento

A página que lista os documentos que estão anexados ao projeto no sistema (Figura 59) é possível visualizar todos os documentos do projeto. Nesta página é possível incluir, excluir e acessar documentos que estão anexados ao sistema.

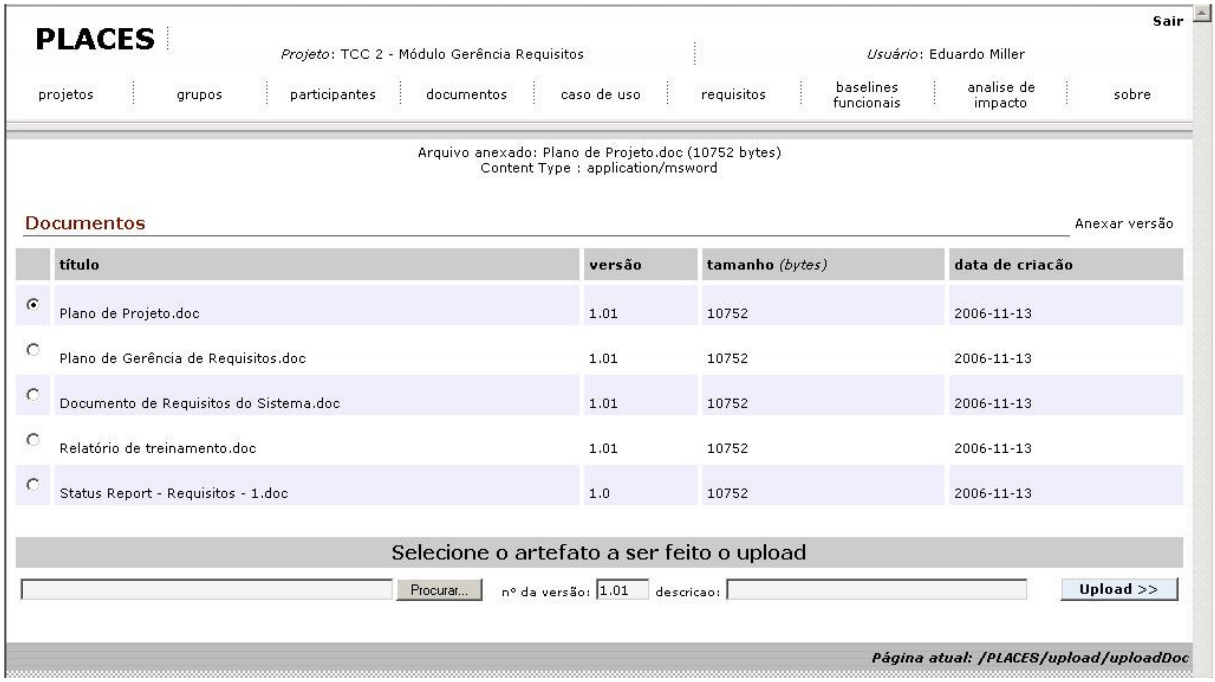


Figura 70 - Lista de documentos anexos ao projeto.

Para adicionar um documento ao projeto, o usuário deve localizar o documento (), preencher o campo "*número da versão*" do documento e colocar uma breve descrição do documento no campo "*descrição*". Para concluir o processo, o botão "Upload" deve ser pressionado.

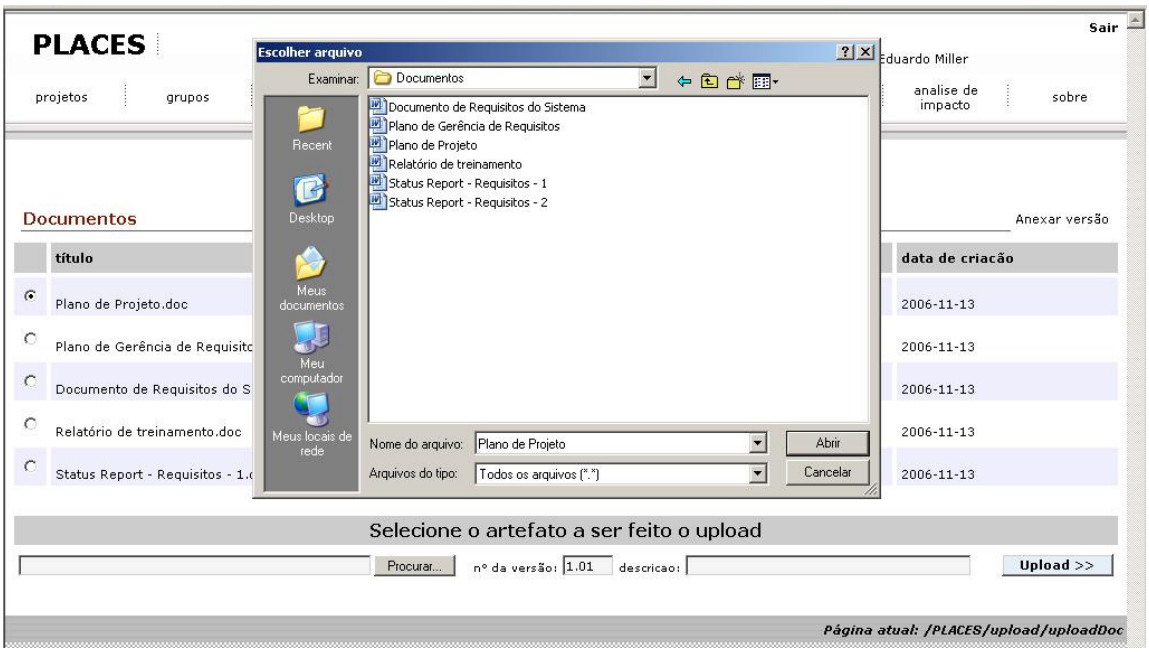


Figura 71 - Adicionar um documento anexo ao projeto.

Para acessar um documento, basta clicar sobre o título do documento. Se o documento puder ser aberto dentro do navegador utilizado, será aberta uma nova janela mostrando o documento. Se o documento não puder ser aberto no navegador ele será, automaticamente, disponibilizado para download.

6.2.4.2 Documentação da opção Anexar Documentos

Nesta seção são detalhados os casos de uso da opção anexar documentos do PLACES. A opção anexar documentos realiza dois casos de uso: o caso de uso de anexar documentos ao projeto e o caso de uso de visualização de um documento anexado.

Caso de Uso 001 - Nome: **Anexar documentos ao projeto**	
Ator ativo: Analista de Sistema	**Atores passivos**: N\A.
Objetivo: Anexar qualquer documento ao projeto	
Pré-condição: O analista de sistema deve estar logado no sistema.	
FLUXO BASE 1. O sistema exibe a lista de documento (Figura 70). 2. O analista seleciona a opção localizar. 3. O sistema apresenta a tela "escolher arquivo"(Figura 71). 4. O analista escolhe o arquivo e pressiona OK. 5. O analista preenche os dados do documento e opta por salvá-lo. 6. O sistema grava os dados do arquivo, faz upload do arquivo e mostra a lista de documentos do projeto. **Pós-condição**: A baseline está cadastrada no sistema.	
FLUXO ALTERNATIVO – FA-01 Cancelamento 1. O sistema sai da tela de cadastro de baselines. **Pós-condição**: Nenhuma alteração é feita no sistema.	
FLUXO DE EXCEÇÃO – FE-1 Erro ao salvar documento No passo 6 do fluxo base, o sistema não consegue salvar o documento. 1. O sistema sai da tela de cadastro da baseline e apresenta uma tela de erro. **Pós-condição**: Nenhuma alteração no sistema	

Tabela 18 - Caso de Uso da opção anexar documento.

A Figura 72 mostra o diagrama de seqüência do caso de uso da geração e exibição da matriz de rastreabilidade do PLACES.

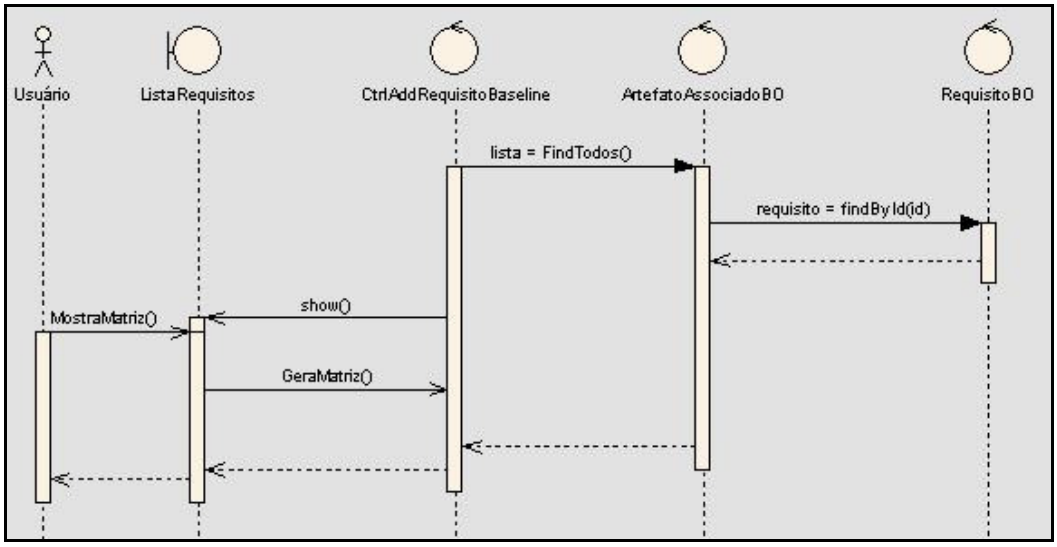


Figura 72 - Diagrama de Seqüência Anexar Documentos

6.3 TECNOLOGIA UTILIZADA

Para o desenvolvimento da ferramenta web foi utilizada a linguagem de programação JAVA, sendo que as páginas web foram escritas em JSP (*Java Server Pages*). O desenvolvimento seguiu o padrão MVC (*Model-View-Controller*) (SUN, 2005). Este padrão separa o código em apresentação, controle e modelo de dados. No PLACES esta divisão foi feita principalmente em três pacotes, *WebContent* (*view*), *com.business.delegate* (*controller*), *com.transfer.object* (*model*) (Figura 73).

Foram utilizados os *frameworks Spring* (RAIBLE, 2005) e o *Hibernate* (BAUER; KING, 2005). O *Spring* é utilizado para controlar a comunicação entre a camada de apresentação (*view*) e a camada de controle (*controller*), permitindo a passagem de objetos (instâncias de classes) de uma pagina para outra. O *Hibernate* é um *framework* para a camada de persistência, de acesso ao banco de dados, fazendo o mapeamento objeto-relacional. Nas próximas seções será descrito o funcionamento desses *frameworks*.

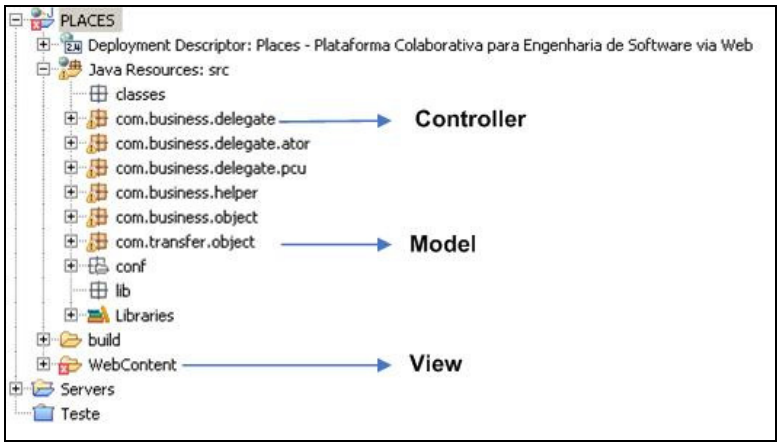


Figura 73 - Divisão MVC no PLACES.

6.3.1 Ferramentas Utilizadas

Para o desenvolvimento da ferramenta web foram utilizadas várias ferramentas, dentre as quais foram eleitas algumas, consideradas mais importantes para citar:

Plataforma Eclipse 3.2.1: esta ferramenta foi utilizada como ambiente de desenvolvimento para o desenvolvimento do módulo de gerência de requisitos, objeto de estudo deste trabalho. O Eclipse é uma ferramenta de código aberto que fornece um ambiente para desenvolvimento de aplicações WEB em Java. Os detalhes para obtenção e instalação da ferramenta serão descritos em anexo a este trabalho.

TomCat 5.5.17: O TomCat foi utilizado como servidor de aplicação para a execução da aplicação WEB. Os detalhes para obtenção e instalação da ferramenta estão descritos em anexo a este trabalho.

MySQL 5.1: banco de dados utilizado para armazenamento de informações do PLACES. Os detalhes para obtenção e instalação da ferramenta estão descritos em anexo a este trabalho.

6.3.2 O *framework* SPRING

O *Spring* é um *framework* de código aberto (WALLS; BREIDENBACH, 2005) que possibilita o uso de *JavaBeans* para executar objetos parecido com a tecnologia EJB (*Enterprise JavaBeans*)(SUN, 2005). O *Spring* permite a simplicidade, testabilidade e baixo acoplamento do código (WALLS; BREIDENBACH, 2005, p.7). O *Spring* fica localizado na camada de negócios no modelo MVC, gerenciando os objetos de negócio, transações, sessões do *Hibernate* entre outras funções.

Este *framework* reduz o acoplamento, porque o *Spring* utiliza um arquivo de configuração para mapear as dependências entre os objetos. Desse modo, cada objeto pode ser escrito independente de outro, utilizando uma técnica conhecida como *inversion of control* (IoC) (WALLS; BREIDENBACH, 2005, p.8).

O arquivo de configuração do *Spring* é divido em duas partes (quadro 2), a primeira parte do arquivo é composto dos mapeamentos da localização das classes controladoras (*controller*) do PLACES para um identificador, a segunda parte relaciona esse identificador com uma chave (*key*). Essa chave é usado pelo PLACES como uma chave de localização dos controladores.

Testabilidade é uma vantagem na utilização do *Spring*. Isso ocorre pois o *Spring* reduz o acoplamento do código. Sendo assim não é preciso ter o programa todo pronto para testar uma classe, as classes podem ser testadas conforme são desenvolvidas. A Tabela 19mostra um extrato do arquivo de configuração do PLACES (*placesapp-servlet.xml*).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN""http://www.springframework.org/dtd/spring-beans.dtd">
<!-- Mapeamentos do Framework SPRING -->


<beans>      <!-- Beans Controladores do PLACES-->
<bean id="controladorDecasodeuso" class="com.business.delegate.ControladorCasoDeUso"/>
<bean id="controladoraddDecasodeuso" class="com.business.delegate.ControladorAddArtefato"/>
<bean id="controladorlistaartefato" class="com.business.delegate.ControladorListaArtefatos"/>
<bean id="controladorsalvaedicaoartefato" <bean id="controladorconfigurarfatorajuste"
class="com.business.delegate.pcu.ControladorConfigurarFatorAjuste"/>
<bean id="controladorsalvaedicaofatorajuste"
class="com.business.delegate.pcu.ControladorSalvaEdicaoFatorAjuste"/>
<bean id="controladoraddfatorajuste" class="com.business.delegate.pcu.ControladorAddFatorAjuste"/>
<bean id="controladorsalvafatorajuste" class="com.business.delegate.pcu.ControladorSalvaFatorAjuste"/>
<bean id="controladordeletefatorajuste" class="com.business.delegate.pcu.ControladorDeleteFatorAjuste"/>
<bean id="controladorEditProjeto" class="com.business.delegate.ControladorEditProjeto"/>
<bean id="controladorcalculapontoscasodeuso"

     <!-- Mapeamento dos controladores -->
    <bean id="urlMapping" class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
        <property name="mappings">
            <props>
                <!-- Controlador de Casos de Uso -->
                <prop key="/addArtefato.exec">controladorDecasodeuso</prop>
                <prop key="/salvaartefato.exec">controladoraddDecasodeuso</prop>
                <prop key="/editaartefato.exec">controladoreditaartefatp</prop>
                <prop key="/salvaedicao.exec">controladorsalvaedicaoartefato</prop>
                <!-- Controlador de Pontos de Casos de Uso -->
             <prop key="/configurarfatordeajuste.exec">controladorconfigurarfatorajuste</prop>
             <prop key="/editaFA.exec">controladorsalvaedicaofatorajuste</prop>
              <!-- Cadastro Atores do Sistema -->
              <prop key="/salvaator.exec">controladorsalvaator</prop>
              <prop key="/removeAtor.exec">controladorapagaator</prop>
            </props>
        </property>
    </bean>
<bean id="messageSource" class="org.springframework.context.support.ResourceBundleMessageSource">
        <property name="basename"><value>messages</value></property></bean>
```

Tabela 19 - Arquivo de configuração do Spring.

Este *framework* permite o desenvolvimento de aplicações complexas a partir de componentes simples. No *Spring*, as classes utilizadas na aplicação são declaradas em um arquivo XML (*placesapp-servlet.xml*, ver Tabela 19). O *Spring* também contém uma infra-estrutura de funcionalidades (gerenciamento de transação, persistência por meio de framework integração, entre outros) deixando a lógica do aplicativo para o desenvolvedor (WALLS; BREIDENBACH, 2005, p.7). Assim resta ao desenvolvedor escrever o código especifico para cada classe. Além de o Spring controlar as transações ele controla a interligação entre as camadas do modelo MVC. Para isso o *framework* utiliza os pacotes *Controller* e *ModelAndView* . Neste trabalho esses pacotes são importados pela classe *ControladorAction* (tabela 14).

```
package com.business.delegate;

import javax.servlet.*;
import javax.servlet.http.*;
…
import org.springframework.web.servlet.mvc.Controller;
import org.springframework.web.servlet.ModelAndView;


public class ControladorAction implements Controller{
…
}
```

Tabela 20 - Classe ControladorAction.

6.3.3 O *framework Hibernate*

O *Hibernate* (BAUER; KING, 2005) possibilita o desacoplamento entre a camada de negócio e a camada de persistência ao mesmo tempo em que faz o mapeamento objeto-relacional. Ele permite que os dados persistidos passem para a camada de negócio como objetos, seguindo a estilo de programação orientado a objetos (BAUER; KING, 2005, p.2).

O *Hibernate* possibilita que o DBA (administrador de banco de dados) e programador possam trabalhar de forma independente. O programador não precisa saber exatamente a estrutura das tabelas no banco de dados, mas apenas saber qual objeto representa os dados de que ele precisa. O programador também não precisa conhecer a linguagem SQL (*Structured Query Language*) e suas variações de acordo com o banco de dados. O DBA, por outro lado, é quem vai escrever o código das consultas, inserções, alterações e de exclusões no banco de dados de acordo com a linguagem HQL (*Hibernate Query Language*) (BAUER; KING, 2005, p.23).

No arquivo *Hibernate.properties* é o arquivo de configuração que será utilizado pelo *Hibernate* para poder acessar o banco de dados. Neste arquivo está contido caminho do *driver* (conjunto de arquivos necessários para se conectar com o banco de dados) de conexão com o banco de dados, o local onde se encontra instalado o banco de dados, o nome do banco de dados o qual se quer conectar, o *username* e a senha. No caso do PLACES o banco de dados utilizados é o *MySQL*, o nome do banco é places, conforme pode ser observado na Tabela 21.

```
hibernate.query.imports net.sf.hibernate.test, net.sf.hibernate.eg
#################
### Platforms ###
#################
## MySQL
hibernate.dialect net.sf.hibernate.dialect.MySQLDialect
hibernate.connection.driver_class com.mysql.jdbc.Driver
hibernate.connection.driver_class org.gjt.mm.mysql.Driver
hibernate.connection.url jdbc:mysql://127.0.0.1:3306/places?autoReconnect=true
hibernate.connection.username root
hibernate.connection.password *****
```

Tabela 21 - Arquivo de configuração do hibernate.

O HQL é uma linguagem similar ao SQL, porém muito mais simples. Ou seja, quem conhece a linguagem SQL não terá problemas com a HQL, pois ela é uma linguagem SQL voltada para a orientação objeto (BAUER; KING, 2005, p.33). Na Tabela 22 há um exemplo da linguagem HQL, notar que na linguagem SQL seria necessário o comando *select* para fazer uma consulta

no banco, na HQL, este comando é omitido, além disso ao invés de colocar o nome da tabela a qual se deseja fazer a consulta, é colocado o nome da classe que representa um objeto (o qual pode representar uma ou mais tabelas no banco de dados). O *Hibernate* já monta o objeto na consulta, não sendo necessário montá-lo.

Assim como o comando *select*, os comandos *insert*, *update* e *delete* do SQL são omitidos pelo *Hibernate*. Note que com isso o programador não precisa saber dos relacionamentos das tabelas do banco de dados, apenas saber quais objetos ele precisa para desenvolver o programa.

```
public FatorAjusteTO findByKey(int key) throws Exception {
        Session session = HibernateUtil.currentSession();
        List fatores = session.find("from FatorAjusteTO as f where f.idFatorAjuste= "+key);
        FatorAjusteTO fto = (FatorAjusteTO)fatores.get(0);
        return fto;
    }
```

Tabela 22 - Exemplo de HQL.

Neste trabalho, a maior dificuldade com o *Hibernate* foi sua configuração inicial, pois ela exige o conhecimento das *tags* XML (*extensible markup language*) específicas ao *Hibernate*. Além de aprender a utilizar e configurar o *Hibernate*, foi também necessário conhecer a linguagem XML (BAUER; KING, 2005, p.34).

O mapeamento objeto-relacional é feito por meio do arquivo XML. O arquivo *hibernate.cfg.xml* é o arquivo de configuração do *Hibernate*, por meio desse arquivo o *Hibernate* mapea quais arquivos representam os mapeamentos objeto-relacional (ver Tabela 23).

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE    hibernate-configuration    PUBLIC    "-//Hibernate/Hibernate    Configuration    DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-2.0.dtd">
<hibernate-configuration>
    <session-factory>
                <!-- Arquivos de mapeamento -->
        <mapping resource="ArtefatoGrupoTO.hbm.xml"/>
        <mapping resource="ArtefatoTO.hbm.xml"/>
        <mapping resource="ParticipanteTO.hbm.xml"/>
        <mapping resource="TipoArtefatoTO.hbm.xml"/>
        <mapping resource="FatorDeAjusteTO.hbm.xml"/>
        <mapping resource="AtorTO.hbm.xml"/>
        <mapping resource="PCUTO.hbm.xml"/>
     </session-factory>
</hibernate-configuration>
```

Tabela 23 - Arquivo hibernate.cfg.xml.

Os arquivos mapeados no arquivo *hibernate.cfg.xml* são os arquivos que representam a relação entre uma tabela e um objeto, os campos das tabelas são mapeados para os atributos do objeto conforme quadro 7. Neste caso o objeto ArtefatoTO esta relacionado com a tabela

Artefato (esta tabela pode representar um arquivo tipo documento do projeto, um caso de uso, um requisito funcional, uma regra de negócio entre outros) no banco de dados.

Note que nesse arquivo é colocado qual a tabela do banco de dados será mapeada, no caso a tabela Artefato, e o objeto o qual representará essa tabela, neste caso ArtefatoTO que está localizado no pacote *com.transfer.object*.

Há um campo identificador na tabela Artefato, neste caso ID_ARTEFATO, o qual é do tipo *auto_increment*, ou seja, a cada nova inserção no banco o valor dessa coluna é incrementado. Isso é necessário para que o *Hibernate* possa inserir novos dados na tabela do banco de dados. Para mapear um atributo com a coluna da tabela utiliza-se a *tag property*. Nesta *tag* o campo *name* representa o atributo da classe, o campo *type* representa o tipo do atributo da classe, o campo *column* representa o nome da coluna da tabela do banco de dados, e por fim o campo *length* representa o tamanho de cada dado da coluna da Tabela 24.

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-2.0.dtd" >
<hibernate-mapping>
<!--
    Created by the Middlegen Hibernate plugin 2.2

    http://boss.bekk.no/boss/middlegen/
    http://www.hibernate.org/
-->
<class
    name="com.transfer.object.ArtefatoAssociadoTO"
    table="ARTEFATO_ASSOCIADO"
>
    <id
        name="idArtefatoAssociado"
        type="java.lang.Integer"
        column="ID_ARTEFATO_ASSOCIADO"
    >
                    <meta attribute="field-description" inherit="false">
                       auto_increment
                    </meta>

        <generator class="native" />
    </id>
    <property
        name="idTipoAssociacao"
        type="int"
        column="ID_TIPO_ASSOCIACAO"
        not-null="false"
        length="11"
    />
    <property
        name="idArtefatoLigado"
        type="int"
        column="ID_ARTEFATO_LIGADO"
        not-null="false"
        length="11"
    />
    <property
        name="idArtefatoAtual"
        type="int"
        column="ID_ARTEFATO_ATUAL"
        not-null="false"
        length="11"
      />
    <property
        name="textoAdicional"
        type="java.lang.String"
        column="TEXTO_ADICIONAL"
        not-null="false"
        length="65535"
    />
    <!-- Associations -->
</class>
</hibernate-mapping>
```

Tabela 24 - Arquivo de mapeamento objeto-relacional do *hibernate.*

6.4 PADRÕES DE PROJETO ADOTADOS

O desenvolvimento do módulo de estimativa de software seguiu a arquitetura MVC (*Model-view-Controller*) (SUN, 2005), onde o código é divido em três camadas. A camada *model* que representa o modelo dos dados na aplicação. A camada *view* que representa a parte da interação do sistema com o usuário e a camada *controller* que representa a parte da lógica da aplicação, fornecendo serviços comuns como autorização e gerenciamento de erros (HUSTED *et al*, 2004, p.580). Neste trabalho, a parte *view* é parte do pacote *webcontent*, a parte *model* é implementada no pacote *transfer.object* e a parte do *controller* é implementada no pacote *business.object*.

Para implementar o módulo de gerência de requisitos, com base na arquitetura MVC, foram utilizados alguns padrões de projeto conhecidos, tais como *Front Controller*, *View Helper*, *Composite View*, *Data Access Model*, *Business Delegate*, *Business Object* e *Transfer Object* (ou *Value Object*). Nas próximas seções, estes padrões são brevemente apresentados e seu uso no desenvolvimento é comentado.

6.4.1 Front Controller

O *Front Controller* "fornece um controlador centralizado para gerenciar o tratamento de uma solicitação" (ALUR, Deepak et al, 2002, p.116). É um *container* para manter a lógica de processamento comum que ocorre na camada de apresentação e que, por outro lado, é colocado de forma errônea em uma visualização. Um controlador trata as solicitações e gerencia a recuperação de conteúdo, segurança, gerenciamento e navegação da visualização, delegando ao componente *Dispatcher* (ver seção 6.4.8) a função de distribuir para uma visualização (ALUR *et al*, 2002, p.125). Esse padrão permite a centralização do código relacionado com os serviços do sistema, serviços de segurança, recuperação do conteúdo, gerenciamento da exibição e navegação, portanto esse código da aplicação não é duplicado infinitamente ou misturado com o conteúdo da exibição (HUSTED *at al*, 2004, p.553).

No PLACES o *Front Controller* é implementado na classe "ControladorAction". O *Dispatcher* (ver seção 6.4.8) é configurado no arquivo *placesapp-servlet.xml*, o qual é utilizado pelo *framework Spring*.

6.4.2 View Helper

O *View Helper* encapsula a lógica que não esteja relacionada à formatação da apresentação em componentes auxiliares (*Helper*) (ALUR *et al*, 2002, p.116). O *View Helper* encoraja a separação do código relacionado à formatação de outra lógica de negócios. Sugere utilizar os componentes *helper* para encapsular a lógica relacionada ao início da recuperação do conteúdo, validação e adaptação e formatação do modelo. O componente *view* é, depois, deixado para encapsular a formatação da apresentação.

```
/* getXMLValue.java
 * Created on 25 de Abril de 2005, 18:48  */
package com.business.helper;
import javax.naming.*;

public class GetXMLValueUtil {
    private Context env;
    private String XMLValue;
    /** Creates a new instance of getXMLValue */
    public GetXMLValueUtil() {
        try{
            env = (Context) new InitialContext().lookup("java:comp/env");
        }
        catch(javax.naming.NamingException e){
            System.out.println("Erro ao ler do arquivo web.xml...");
        }
    }
    /** Retorna o valor da tag xml com nome appName
      * @return       */
     public String getXMLErroLogin(){
        try{
            //Pega o conteï¿½do de uma tag XML para a memï¿½ria primï¿½ria.
            XMLValue = (String) env.lookup("erroLogin");
        }
        catch(javax.naming.NamingException erro){
                System.out.println("Erro ao ler o web.xml...\n:<br>"+erro);
                XMLValue = "";
        }
        return XMLValue;
    }    /**
      * Retorna o valor da tag xml com nome appName
      * @return       */
     public String getXMLAppName(){
        try{
            //Pega o conteï¿½do de uma tag XML para a memï¿½ria primï¿½ria.
            XMLValue = (String) env.lookup("appName");
        }
        catch(javax.naming.NamingException erro){
                System.out.println("Erro ao ler o web.xml...\n:<br>"+erro);
                XMLValue = "";
        }
        return XMLValue;
    }
    /**     *
     * @return      */
    public String getXMLAppPath(){
        try{
            //Pega o conteï¿½do de uma tag XML para a memï¿½ria primï¿½ria.
            XMLValue = (String) env.lookup("path");
        }
        catch(javax.naming.NamingException erro){
                System.out.println("Erro ao ler o web.xml...\n:<br>"+erro);
                XMLValue = "";
        }
        return XMLValue;
    }
}
```

Tabela 25 - Classe View Helper.

O *View Helper* é implementado no pacote "*com.business.helper*" no PLACES. A Tabela 25 mostra a classe "*GetXMLValueUtil.java"* que é utilizado entre outras funções para retornar uma mensagem de erro no caso de uma tentativa de autenticação com o login do usuário ou a senha esteja errados.

6.4.3 Composite View

O *Composite View* cria uma visualização (*view*) agregada a partir de subcomponentes muitos pequenos (ALUR, Deepak et al, 2002, p.116). O *Composite View* sugere compor uma visualização de inúmeras parcelas pequenas. Este padrão é implementando por meio das

taglibs que compõe a tela principal. No PLACES para compor a visualização da página *web addRequisito.jsp* (Tabela 27), foi utilizado a tag *exibeComplexidadeRequisito.tag* (Tabela 26).

```
<%@tag description="Exibe a complexidade do requisito" pageEncoding="UTF-8"%>
<select name="complexidade">
    <option value="1" selected>baixa</option>
    <option value="2">média</option>
    <option value="3">alta</option>
</select>
```

Tabela 26 - Arquivo exibeComplexidadeRequisito.tag.

```
<%@taglib prefix="tag" tagdir="/WEB-INF/tags"%>

<form name="form1" method="post" action="salvarequisito.exec">
    <table border="0" cellpadding="4" cellspacing="4" width="100%"  >
    <tbody>
        <tr>
           .
           .
           .
        </tr>
        <tr>
        <td>
        <table >
             .
             .
             .
            <tr>
                <td width="12%" align="left" bgcolor='#eeeefc'><b>complexidade</b></td>
                <td width="42%" align="left" bgcolor='#eeeefc'><b> tipo requisito</b></td>
                <td width="46%" align="left" bgcolor='#eeeefc'><b>palavras-chaves</b></td>
            </tr>
            <tr>
                <td width="12%" align="left" bgcolor='#eeeefc'>
                    <tag:exibeComplexidadeRequisito/>
                </td>
                <td width="42%" align="left" bgcolor='#eeeefc'>
                    <tag:exibeTipoRequisito/>
                </td>
                <td width="46%" align="left" bgcolor='#eeeefc'>
                    <input name="palavrachave" id="palavrachave" type="text" width="400" value="">
                </td>
            </tr>
```

Tabela 27 - Extrato da página addRequisito.jsp.

Neste caso foram omitidas algumas linhas de código html na página *addRequisito.jsp* por meio da *tag exibeComplexidadeRequisito.tag* possibilitando um código html limpo. Em outras páginas web foram retirados os códigos java das paginas JSP, separando o código dinâmico do estático. Esta técnica foi utilizada em todo código.

6.4.4 Business Delegate

O *Business Delegate* decompõe as camadas de apresentação e serviço e fornece uma *interface* de *facade* (fachada de acesso) e de *proxy* aos serviços (ALUR *et al*, 2002, p.116). O *Business Delegate* reduz o acoplamento entre as camadas e fornece um ponto de entrada para acessar os serviços que são fornecidos por outra camada. O *delegate* pode também fornece armazenamento em *cache* de resultados para as solicitações comuns para melhorar o

desempenho. O padrão *Business Delegate* é implementado no pacote "*com.business.delegate*" e seus subpacotes.

No quadro 11, um exemplo de classe utilizando o padrão *Business Delegate*, a qual recebe uma requisição e redireciona para a página correspondente no caso a página *addRequisito.jsp* (Tabela 27).

```
package com.business.delegate;

import java.util.ArrayList;
import javax.servlet.http.*;
import org.springframework.web.servlet.ModelAndView;
import com.business.object.ParticipanteBO;
import com.transfer.object.ProjetoTO;

public class ControladorRequisito extends ControladorAction {
  public ModelAndView handleRequest(HttpServletRequest request, HttpServletResponse response) throws
Exception {
        super.processRequest(request,response);

        String url = "main.jsp?link=requisito/addRequisito&autor="+super.pto.getNome();
        super.logger.debug("ControladorRequisito: navegando para " + url);
        super.setView(url);
        try
          {
          // dados do projeto e participantes
          ParticipanteBO partbo = new ParticipanteBO();
          ProjetoTO projto = (ProjetoTO)super.sessionParticipante.getAttribute("projeto");
          ArrayList                                listaParticipantes                                =
(ArrayList)partbo.findByIdProjeto(projto.getIdProjeto().toString());
          super.logger.info("Projeto:" + projto); // Loga o projeto to
          super.logger.info(listaParticipantes); // Loga a lista
          super.sessionParticipante.setAttribute("listaParticipantes",listaParticipantes);
          return super.handleRequest(request, response);
          }
        catch(java.lang.Exception e)
          {
          super.logger.error(super.pto.getIdParticipante().toString());
          super.logger.equals(e);
          return new ModelAndView(url);
          }
    }
}
```

Tabela 28 - Classe ControladorRequisito.java.

6.4.5 Data Access Object

O *Data Access Object* (DAO) é padrão utilizado para a persistência dos dados. Fornece acesso transparente aos dados (ALUR *et al*, 2002, p.116). O *Data Access Object* fornece baixo acoplamento entre as camadas de negócio e de persistência. O *Data Access Object* intercepta e presta serviços a todos os acessos à camada de persistência, tornando os detalhes da implementação das camadas de persistência transparentes para os clientes. Os dados na camada de persistência podem persistir em sistemas de bancos de dados, sistemas proprietários, outros sistemas e serviços externos.

No PLACES utiliza esse padrão por meio do *Hibernate* (ver seção 6.3.3). Neste caso utilizando as classes do tipo *Transfer Object* (ver seção 6.4.7). Estas classes representam uma ou mais tabelas no banco de dados. Essas classes são mapeadas pelo *Hibernate*. Utilizam-se

as classes do tipo *Business Object* para ter acesso as classes do tipo *Transfer Object.* Desta forma, é utilizado o padrão DAO por meio da configuração dos arquivos utilizados pelo *Hibernate*, a criação das classes dos tipos *Transfer Object* e *Business Object*.

6.4.6 Business Object

O *Business Object* (SUN, 2005) um objeto que implementa a lógica de negócios e/ou os dados de negócios. Os dados de negócio e a lógica de negócios são implementados em objetos denominados objetos de negócios (*business object*). No J2EE, os objetos de negócio são implementados como *beans* de sessão (*session beans*) ou de entidade (SUN, 2005). Este padrão foi implementando por meio do *framework Hibernate*.

No PLACES as classes *Business Object* estão no pacote *com.business.delegate*. Essas classes são utilizadas no PLACES para fazer a inserção, a consulta, a atualização e apagar os dados do banco de dados. Cada classe do tipo *Business Object* tem uma classe do tipo *Transfer Object* correspondente.

6.4.7 Transfer Object

O *Transfer Object* (objeto de dados) um objeto Java arbitrário que é utilizado para transportar dados de um objeto/camada para outro. Normalmente não contém nenhum método de negócios. Pode ser projetado com atributos públicos ou fornecido com métodos *get's* e *set's* para obter e atribuir valores dos atributos. (ALUR *et al*, 2002, p.125). O *Transfer Object* fornece as melhores técnicas e estratégias para trocar dados através de camadas (isto é, através de limites de sistemas). Esse padrão tenta reduzir a sobrecarga de rede minimizando diversas camadas de rede para receber dados da camada de negócios.

```
package com.transfer.object;


import java.io.Serializable;
import java.util.Date;

/** @author Hibernate CodeGenerator */
public class ArtefatoTO implements Serializable {

    /** identifier field */
    private Integer idArtefato;

    /** nullable persistent field */
    private Integer idParticipanteCriou;

    /** nullable persistent field */
    private Integer idTipoArtefato;

    /** nullable persistent field */
    private String palavraChave;

    /** nullable persistent field */
    private String codigoArtefato;

    /** nullable persistent field */
    private String nomeArtefato;

    /** nullable persistent field */
    private String descricaoArtefato;

    /** nullable persistent field */
    private String grauComplexidade;

    /** nullable persistent field */
    private byte[] documento;
```

Tabela 29 – Implementação Classe ArtefatoTO.java.

Este padrão foi implementado no pacote "*com.transfer.object*" no PLACES, contendo neste pacote todas as classes de dados do PLACES. Todas as classes *Transfer Object* são utilizadas para enviar os dados de uma página para um controlador ou de um controlador para uma página web. As classes *Business Object* (ver seção 6.4.6) são responsáveis recuperar as classes *Transfer Object* do banco de dados. Neste caso utilizando o *Hibernate* (ver seção 4.1.2). Por exemplo, a classe *ArtefatoTO.java* é mapeado em um arquivo XML .

6.4.8 Dispatcher

O *Dispatcher* é um controlador que gerencia a seleção de uma visualização apropriada. Separando esse comportamento do controlador principal (ALUR *et al*, 2002, p.119). Este padrão é implementado por meio do *framework Spring* (ver seção 6.3.2 e Tabela 19). O qual utiliza um arquivo XML (ver Tabela 19) para configurar esse mecanismo.

6.5 RE-AVALIAÇÃO DO PLACES

6.5.1 Avaliação do Módulo de Gerência de Requisitos

Para atender ao objetivo deste trabalho, a mesma avaliação, aplicada na primeira etapa, foi reaplicada no PLACES com o objetivo de medir sua aderência à área de processo de Gerência de Requisitos do CMMI.

A avaliação foi feita sobre a plataforma PLACES na versão 1.0.1.2, gerada como resultado deste trabalho de conclusão. O processo de instalação do ambiente de desenvolvimento e da ferramenta em si foi colocado como anexo a este trabalho em função de ser um processo extenso. Para a avaliação foi utilizado o projeto exemplo disponível no banco de dados (em sua versão 1.3) disponibilizado para a aplicação. Para melhor análise da ferramenta, foi cadastrado um conjunto de requisitos de forma a auxiliar na avaliação executada.

Como referência para aprendizado da ferramenta utilizou-se os trabalhos de conclusão de Moretto (2005) e Souza (2005) que desenvolveram a ferramenta.

A tabela 12 mostra os resultados da avaliação desta ferramenta e registra as evidências que foram encontradas para suprir as práticas indicadas na PA Gerência de Requisitos do CMMI. Da mesma forma como foi feito na avaliação das demais ferramentas, a coluna "Documentos" foi deixada na planilha para que não se tenha uma planilha específica para a avaliação de cada ferramenta, apesar da versão atual do PLACES não fornecer *templates* para geração de documentos.

Gerenciamento de Requisitos – REQM

Ferramenta: PLACES – Módulo de Gerência de Requisitos

Status		Praticas / Anotações	Documento(s)	Comentários		Direta	Indireta	Pontuação
	SG1	**Os requisitos são gerenciados e as inconsistências com o plano de projeto e produtos de trabalho são identificadas**						**I**
	SP1.1-1	**Obter um entendimento dos Requisitos**						
		A ferramenta oferece cadastramento de casos de uso e requisitos.		Foi criado um cadastro de requisitos para este fim.			**X**	**LI**
		O cadastro de requisitos possui um campo status que indica se o requisito foi aceito, indicando que houve um entendimento sobre o requisito. Este campo pode ser preenchido manualmente pelo usuário.		Existe campo específico no cadastro de requisitos. Foi previsto um fórum de discussões para auxiliar o atendimento desta prática, porém não foi implementado.			**X**	**LI**

	PF							**LI**
	SP1.2-2	**Obter Comprometimento dos participantes do projeto com Requisitos**						
		A ferramenta fornece um campo indicando a quem o requisito foi atribuído. O participante deverá marcar o campo aceite para indicar seu comprometimento com a atribuição.		Existe campo específico no cadastro de requisitos.			**X**	**LI**
	PF							**LI**
	SP1.3-1	**Gerenciar Mudanças de Requisitos a medida que elas ocorram durante o projeto**						
				Esta prática está sendo atendida por outro TCC. Entretanto, depois que o requisito é adicionado ou modificado, os outros recursos da ferramenta irão completar a prática, incluindo o rastreamento do atendimento do requisito.				**NI**
	PF							**NI**
	SP1.4-2	**Manter Rastreabilidade Bidirecional entre os Requisitos e o plano de projeto e demais artefatos do projeto**						
		A ferramenta oferece rastreabilidade horizontal entre os requisitos.		Foi criada a Matriz de Rastreabilidade com base na associação entre os requisitos. Para uma matriz mais completa faltaria associar o requisito com as atividades que o executam.			**X**	**PI**
	PF							**PI**
	SP1.5-1	**Identificar Inconsistências entre o plano do projeto e demais artefatos e os requisitos**						
		Definição de rastreabilidade horizontal entre requisitos.		Foi implementada a associação entre os requisitos.			**X**	**PI**
		A ferramenta oferece uma **matriz de rastreabilidade** entre requisitos.		Foi criada a Matriz de Rastreabilidade com base somente na associação entre os requisitos. Para permitir uma matriz de rastreabilidade entre os requisitos e os demais artefatos do projeto seria necessário implementar este tipo de associação.			**X**	**PI**
	PF							**PI**
	GG2	**Institucionalizar um processo gerenciado**						**I**
	GP2.1	**Estabelecer e manter uma política organizacional para o planejamento e execução do processo de gerenciamento de requisitos.**						
		A ferramenta permite anexar ao projeto o documento para estabelecer e manter uma política organizacional para o planejamento e execução do processo de gerenciamento de requisitos, porém não fornece *templates* para este fim.		Foi corrigida a função de anexar documentos ao projeto para apoiar esta prática.			**X**	**PI**
	PF							**PI**
	GP2.2	**Estabelecer e manter o plano para a execução do processo de gerenciamento de requisitos.**						
		A ferramenta permite anexar ao projeto o documento do Plano de Gerência de Requisitos, porém não fornece *templates* para este fim.		Foi corrigida a função de anexar documentos ao projeto para apoiar esta prática.			**X**	**PI**
	PF							**PI**
	GP2.3	**Fornecer os recursos adequados para a execução do processo de gerenciamento de requisitos, desenvolvimento de produtos de trabalho e fornecimento dos serviços do processo.**						

		A ferramenta fornece recursos parciais para execução do processo, atendendo parcialmente o processo de entendimento comum sobre os requisitos.		Foi corrigida a função de anexar documentos ao projeto para apoiar esta prática.			X	PI
	PF							PI
	GP2.4	**Atribuir responsabilidades e autoridade para a execução do processo, desenvolvimento dos produtos de trabalho e fornecimento dos serviços do processo de gerenciamento de requisitos.**						
		A ferramenta permite anexar ao projeto os documentos de registro de responsabilidades dos stakeholders, porém não fornece *templates* para este fim.		Foi corrigida a função de anexar documentos ao projeto para apoiar esta prática.			X	PI
	PF							PI
	GP2.5	**Treinar as pessoas para executar e suportar o processo de gerenciamento de requisitos, conforme necessário.**						
		A ferramenta permite anexar ao projeto os documentos de registro de treinamento dos stakeholders, porém não fornece *templates* para este fim.		Foi corrigida a função de anexar documentos ao projeto para apoiar esta prática.			X	PI
	PF							PI
	GP2.6	**Colocar os produtos de trabalho definidos do processo de gerenciamento de requisitos sob os níveis apropriados de gerenciamento de configurações.**						
		A ferramenta fornece um cadastro de baselines de requisitos.		Foi criado um cadastro de requisitos.			X	PI
		A ferramenta fornece recurso de versionamento dos requisitos, gravando histórico de alteração dos mesmos.		As alterações nos requisitos são guardadas em uma tabela de histórico.			X	PI
	PF							PI
	GP2.7	**Identificar e envolver os stakeholders relevantes do processo de gerenciamento de requisitos, conforme planejado.**						
		Os integrantes do projeto são identificados através de **cadastro próprio**. No cadastro de requisitos os stakeholders são definidos como responsáveis pela execução dos requisitos.		Foi corrigida a função de anexar documentos ao projeto para apoiar esta prática.			X	PI
	PF							PI
	GP2.8	**Monitorar e controlar o processo de gerenciamento de requisitos contra o plano de execução do processo e tomar a ação corretiva apropriada.**						
		A monitoração é manual. A ferramenta permite anexar ao projeto os documentos de registro de monitoração, porém não fornece *templates* para este fim.		A monitoração é manual e de responsabilidade do usuário. A ferramenta indica porém não dá suporte direto à este recurso.			X	PI
	PF							PI
	GP2.9	**Avaliar objetivamente a aderência do processo de gerenciamento de requisitos contra sua descrição de processo, padrões e procedimentos e tratar as não conformidades.**						
		A avaliação é feita de forma manual e parcialmente apoiada pelo **histórico de alterações** e registrada relatórios de avaliação que são anexados ao projeto.		A avaliação é manual e de responsabilidade do usuário. Foi corrigida a função de anexar documentos ao projeto para apoiar esta prática. A ferramenta oferece um processo parcial a ser seguido que precisaria ser complementado com padrões e procedimentos.			X	PI
	PF							PI

	GP2.10	**Revisar as atividades, status e resultados do processo de gerenciamento de requisitos com o nível mais alto nível de gerência e resolver questões.**						
		A revisão é feita de forma manual e parcialmente apoiada pelo **histórico de alterações, status das baselines** e registrada relatórios de avaliação que são anexados ao projeto.		Foram criados os cadastros de baselines, requisitos e estão sendo gravados os históricos das alterações. Foi corrigida a função de anexar documentos ao projeto para apoiar esta prática.			**X**	**PI**
	PF							**PI**
Resultado geral da avaliação								**I**
Legenda: NI (Não Implementado), PI (Parcialmente Implementado), LI (Largamente Implementado) e FI (Totalmente Implementado)								

Tabela 30 - Planilha de reavaliação da ferramenta PLACES.

Como pode ser observado na Tabela 30, a ferramenta PLACES, nesta segunda avaliação, provê mais suporte ao CMMI que sua versão anterior. O suporte maior está no melhor atendimento das práticas específicas, provido pelo módulo de gerência de requisitos. Um recurso interessante é o recurso de anexar arquivos externos aos projetos. Mesmo que a ferramenta não dê suporte direto à maioria das práticas genéricas e não forneça *templates* para sua execução, é possível atendê-las utilizando qualquer *template* disponível no mercado. A falta da rastreabilidade entre os requisitos e os demais artefatos do projeto (casos de uso, atividades etc) foi decisiva para o atendimento do resultado "I" para a avaliação das práticas específicas.

Mesmo com a evolução obtida, a ferramenta PLACES demonstra ainda que é possível melhorar seus recursos tendo como referência as ferramentas comerciais anteriormente avaliadas.

Desta forma, a avaliação mostrou que, mesmo que a ferramenta PLACES ainda não esteja totalmente alinhada à PA de Gerência de Requisitos do CMMI, há poucos pontos a serem melhorados para apoiar este processo. Como pode ser identificado na planilha de avaliação, a maioria das práticas são atendidas parcialmente e a ferramenta também oferece um recurso para anexar os documentos que apóiam as áreas não atendidas pela ferramenta.

6.5.2 Comparação entre as versões do PLACES

A ferramenta PLACES não possuía, em sua versão inicial, suporte à gerência de requisitos. Com a implementação do Módulo de Gerência de Requisitos, a ferramenta começa a apresentar mais recursos para auxiliar esta área de gerenciamento. Os recursos que mais se destacaram foram o cadastramento dos requisitos, o cadastramento das baselines de requisitos e a matriz de rastreabilidade. Estes três novos recursos implementam, na ferramenta, os

conceitos de requisitos e *baselines*, mantém uma rastreabilidade bidirecional entre estes requisitos e fornece em uma ferramenta para auxiliar a análise de impacto sobre as mudanças ocorridas nos requisitos. Estes novos recursos possibilitam a execução, parcialmente automatizada, das atividades do processo de Gerência de Requisitos. As atividades que não têm suporte direto na ferramenta podem ser controladas através dos documentos anexados ao projeto.

Para fins de comparação entre as versões, foi criada uma tabela mostrando as diferenças, em relação à PA Gerência de Requisitos, entre a versão inicial do PLACES (versão 1.0.1.1) e a versão com o Módulo de Gerência de Requisitos (versão 1.0.1.2). A Tabela 31 faz uma comparação entre as ferramentas analisadas com relação às práticas sugeridas pelo modelo CMMI.

	Praticas da PA Gerência de Requisitos	**Pontuação**	
		PLACES Versão 1.0.1.1	**PLACES Versão 1.0.1.2**
SG1	**Os requisitos são gerenciados e as inconsistências com o plano de projeto e produtos de trabalho são identificadas**	**I**	**I**
SP1.1-1	**Obter um entendimento dos Requisitos**	**PI**	**LI**
SP1.2-2	**Obter Comprometimento dos participantes do projeto com Requisitos**	**NI**	**LI**
SP1.3-1	**Gerenciar Mudanças de Requisitos a medida que elas ocorram durante o projeto**	**NI**	**NI**
SP1.4-2	**Manter Rastreabilidade Bidirecional entre os Requisitos e o plano de projeto e demais artefatos do projeto**	**NI**	**PI**
SP1.5-1	**Identificar Inconsistências entre o plano do projeto e demais artefatos e os requisitos**	**NI**	**PI**
GG2	**Institucionalizar um processo gerenciado**	**I**	**I**
GP2.1	**Estabelecer e manter uma política organizacional para o planejamento e execução do processo de gerenciamento de requisitos.**	**NI**	**PI**
GP2.2	**Estabelecer e manter o plano para a execução do processo de gerenciamento de requisitos.**	**NI**	**PI**
GP2.3	**Fornecer os recursos adequados para a execução do processo de gerenciamento de requisitos, desenvolvimento de produtos de trabalho e fornecimento dos serviços do processo.**	**PI**	**PI**
GP2.4	**Atribuir responsabilidades e autoridade para a execução do processo, desenvolvimento dos produtos de trabalho e fornecimento dos serviços do processo de gerenciamento de requisitos.**	**NI**	**PI**
GP2.5	**Treinar as pessoas para executar e suportar o processo de gerenciamento de requisitos, conforme necessário.**	**NI**	**PI**
GP2.6	**Colocar os produtos de trabalho definidos do processo de gerenciamento de requisitos sob os níveis apropriados de gerenciamento de configurações.**	**NI**	**PI**
GP2.7	**Identificar e envolver os stakeholders relevantes do processo de gerenciamento de requisitos, conforme planejado.**	**NI**	**PI**
GP2.8	**Monitorar e controlar o processo de gerenciamento de requisitos contra o plano de execução do processo e tomar a ação corretiva apropriada.**	**NI**	**PI**

GP2.9	**Avaliar objetivamente a aderência do processo de gerenciamento de requisitos contra sua descrição de processo, padrões e procedimentos e tratar as não conformidades.**	**NI**	**PI**
GP2.10	**Revisar as atividades, status e resultados do processo de gerenciamento de requisitos com o nível mais alto nível de gerência e resolver questões.**	**NI**	**PI**
Legenda: NI (Não Implementado), PI (Parcialmente Implementado), LI (Largamente Implementado) e FI (Totalmente Implementado)			

Tabela 31 - Comparativo entre as versões do PLACES.

Observa-se, pelos dados comparativos mostrados na planilha de avaliação, que apesar de ambas as versões não apoiarem de forma satisfatória a PA Gerência de Requisitos, houve uma evolução considerável da versão 1.0.1.2 em relação aos itens avaliados. Esta evolução é mostrada na Figura 74.

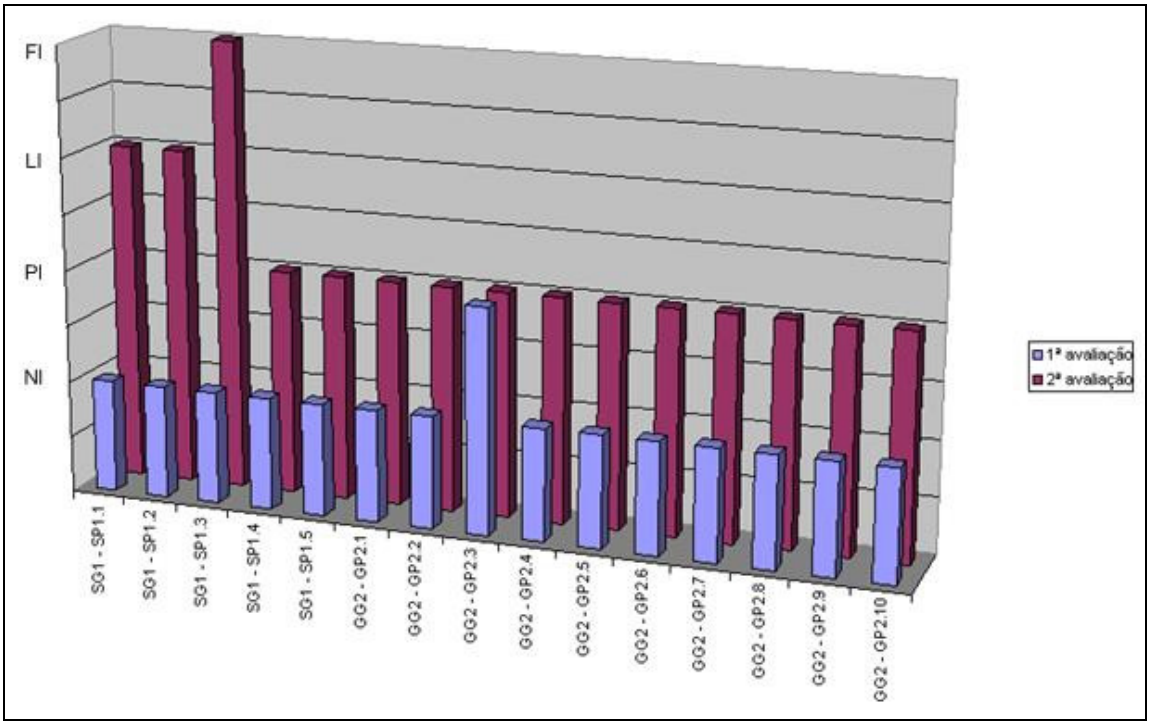


Figura 74 - Comparativo entre versões do PLACES.

As práticas genéricas tiveram uma evolução maior em relação às práticas específicas devido ao recurso de anexar documentos externos ao projeto. Nas práticas específicas, os pontos que melhoraram foram: o entendimento sobre os requisitos, a atribuição de responsabilidades sobre os requisitos, a manutenção de baselines e a matriz de rastreabilidade para avaliação de impacto. A Figura 75 lustra a tela “sobre” com as informações da versão 1.0.1.2.

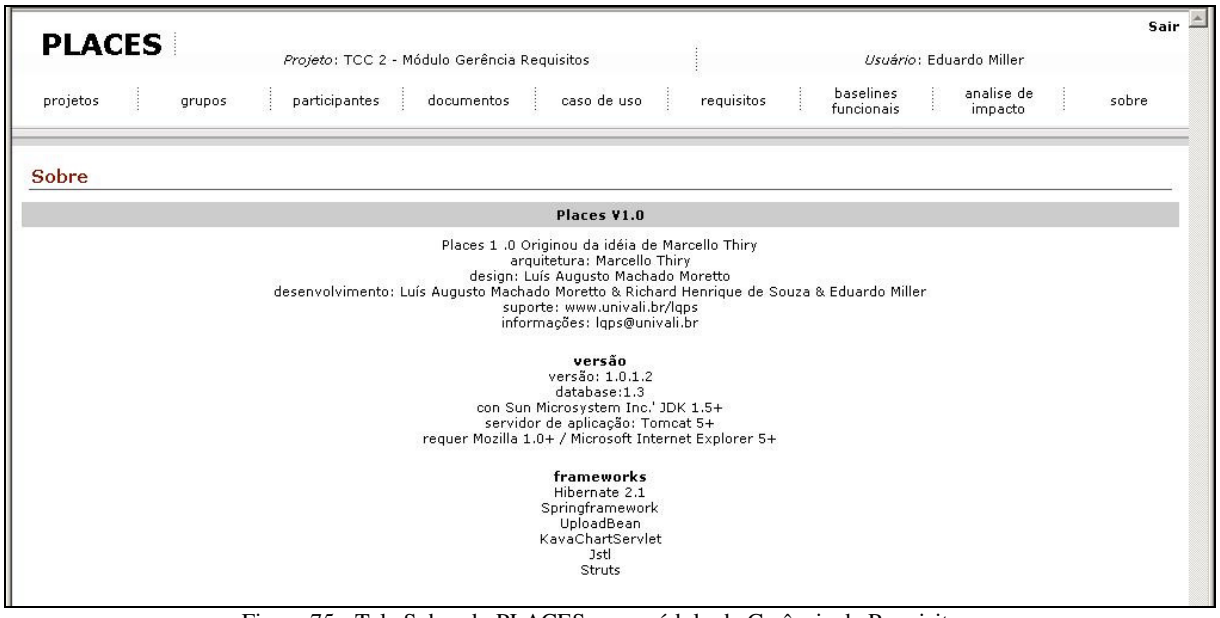


Figura 75 - Tela Sobre do PLACES com módulo de Gerência de Requisitos.

Desta forma, a versão produzida por este trabalho tem condições de apoiar o processo de Gerência de Requisitos e está parcialmente alinhado à PA de Gerência de Requisitos do CMMI.

7 CONCLUSÃO

Este trabalho apresentou, inicialmente, uma revisão do processo de Gerência de Requisitos tendo como referência a Área de Processo de Gerência de Requisitos do CMMI. Com isso conseguiu-se mostrar algumas técnicas para a realização das atividades de gerência de requisitos visando um alinhamento com as práticas definidas pelo modelo. Para contribuir com este estudo, também foram abordados alguns elementos do processo de definição dos requisitos. Desta forma, foi possível ter uma visão mais clara de como o objetivo da área de Gerência de Requisitos pode ser alcançado através do uso das técnicas e artefatos estudados.

Para implementar o Módulo de Gerência de Requisitos do PLACES (objetivo principal do trabalho) foi necessário solidificar uma base de conhecimento sobre Gerência de Requisitos e CMMI. Em seguida, foi efetuada uma avaliação de duas ferramentas comerciais juntamente com a versão do PLACES anterior a este trabalho. A partir do conhecimento adquirido, foi possível estabelecer um método de avaliação para analisar o alinhamento de duas ferramentas comerciais e o PLACES com o modelo CMMI. Com o resultado desta avaliação foi possível verificar que as ferramentas comerciais estão alinhadas com o modelo, sendo opções tecnicamente viáveis para dar suporte ao processo de Gerência de Requisitos segundo o CMMI. Entretanto, o PLACES teve um resultado não satisfatório na avaliação, um resultado esperado uma vez que o PLACES não foi projetado para suportar o processo de Gerência de Requisitos.

O resultado da avaliação gerou uma referência para a construção do novo Módulo de Gerência de Requisitos. O trabalho de desenvolvimento integrou a auto-aprendizagem de técnicas de programação WEB com conceitos do processo de Gerência de Requisitos.

A última etapa do trabalho foi a reavaliação do PLACES, já com seu novo Módulo de Gerência de Requisitos, para identificar a evolução da ferramenta com relação a Área de Processo de Gerência de Requisitos do CMMI.

Este trabalho agregou novas funcionalidades ao ambiente PLACES, tornando-o mais efetivo no suporte ao processo de gerência de requisitos. Os objetos específicos deste trabalho foram alcançados, tais objetivos são descritos a seguir:

- O objetivo de estabelecer um processo de avaliação com base no método oficial do CMMI, o SCAMPI, identificando os critérios para esta avaliação, em conformidade com a área de processo de Gerência de Requisitos do CMMI. Este trabalho descreveu um estudo sobre Gerência de Requisitos e o CMMI, com o objetivo de estabelecer uma referência para a avaliação das ferramentas. O processo de avaliação foi extraído do método de avaliação oficial do CMMI, chamado SCAMPI.
- Aplicar a avaliação em três ferramentas de Gerência de Requisitos, sendo uma delas a plataforma PLACES, verificando seu alinhamento com o modelo CMMI. Este objetivo foi alcançado com a avaliação das ferramentas CaliberRM da Borland, RequisitePro da Rational e a plataforma PLACES desenvolvido no LQPS da Univali. Com esta avaliação foi obtida a referência para o desenvolvimento do Módulo de Gerência de Requisitos do PLACES.
- Estabelecer um plano de melhorias à plataforma PLACES para a criação de um módulo de gerência de requisitos buscando seu alinhamento com esta área de processo do CMMI. Após a avaliação das ferramentas comerciais e da plataforma PLACES, foram listadas melhorias que serviriam de referência para o desenvolvimento do novo módulo do PLACES.
- Desenvolver o módulo de gerência de requisitos proposto para a plataforma PLACES. Foi implementado um módulo de Gerência de Requisitos na plataforma PLACES, buscando como referência a Área de Processo de Gerência de Requisitos do CMMI e as técnicas estudadas nos capítulos 2 e 3.
- Reaplicar a avaliação na plataforma PLACES, comparando os resultados, de forma a avaliar a evolução de seu alinhamento com a área de processo de Gerência de Requisitos. Para alcançar este objetivo, a plataforma PLACES foi reavaliada após a implementação do módulo proposto. Nesta segunda avaliação, foram utilizados os mesmos critérios da primeira avaliação para verificar seu grau de alinhamento com o CMMI.

Como resultado da segunda avaliação, pode ser observada a evolução da ferramenta em seu alinhamento com a área de Gerência de Requisitos do CMMI. Muitas oportunidades de melhoria da versão anterior foram atendidas na versão produzida por este trabalho, tais como um cadastro de requisitos, um cadastro de baselines, rastreabilidade bidirecional entre requisitos e uma matriz de rastreabilidade entre requisitos. Apesar da evolução da plataforma

PLACES, outros recursos podem ser implementados para melhorar seu alinhamento com o modelo CMMI.

Uma vantagem do módulo de gerência de requisitos do PLACES é que esta ferramenta está vinculada à ferramenta de gerência de projetos e à ferramenta de estimativa de tamanho baseada em pontos de caso de uso já implementadas. Outra vantagem é o fato da ferramenta ser acessada via internet.

7.1 TRABALHO FUTURO

Este trabalho pode ser continuado, seguem algumas sugestões para trabalhos futuros:

- Implementar um mecanismo de fórum de discussão para troca de mensagens entre os usuários do sistema, permitindo um melhor esclarecimento sobre os requisitos.
- Permitir a extração de informações históricas da plataforma PLACES, permitindo a comparação entre versões de requisitos, baselines etc.
- Desenvolver um *framework* que para controlar algumas atividades de forma automatizada.
- Desenvolver um sistema de rastreabilidade vertical, permitindo associar requisitos com quaisquer artefatos do projeto (casos de uso, atividades do projeto etc.).
- Fornecer templates para a execução do processo de Gerência de Requisitos.

ANEXO 1 - INSTALAÇÃO DAS FERRAMENTAS

INSTALAÇÃO DAS FERRAMENTAS PARA O DESENVOLVIMENTO E MANUTENÇÃO DO PLACES

1 Ferramentas Utilizadas

1.1 JDK 1.5.0 ou superior. Kit de desenvolvimento JAVA.

1.2 TOMCAT 5.5 ou superior. Servidor de aplicação com suporte a JAVA e JSP.

1.3 MySQL 4.0. Banco de dados a ser utilizado pelo PLACES.

1.4 MyODBC-3.51.11-2-win. Driver de conexão do banco para o sistema operacional Windows.

1.5 Eclipse 3.2 ou superior. Ambiente de desenvolvimento em JAVA.

A seguir são descritos os processos de instalação das ferramentas utilizadas para montar o ambiente de desenvolvimento do PLACES.

2 Instalação do JDK 1.5.0.

O primeiro passo é baixar o pacote de instalação do JDK 5.0 na página da de seu fornecedor, a SUN: http://java.sun.com.

Procure pelo jdk 1.5 (arquivo “jdk-1_5_0_01-windows-i586-p”) para fazer o download.

Após ter baixado o arquivo de instalação do jdk, dar um duplo click no arquivo de instalação (jdk-1_5_0_01-windows-i586-p). A tela de instalação irá aparecer.

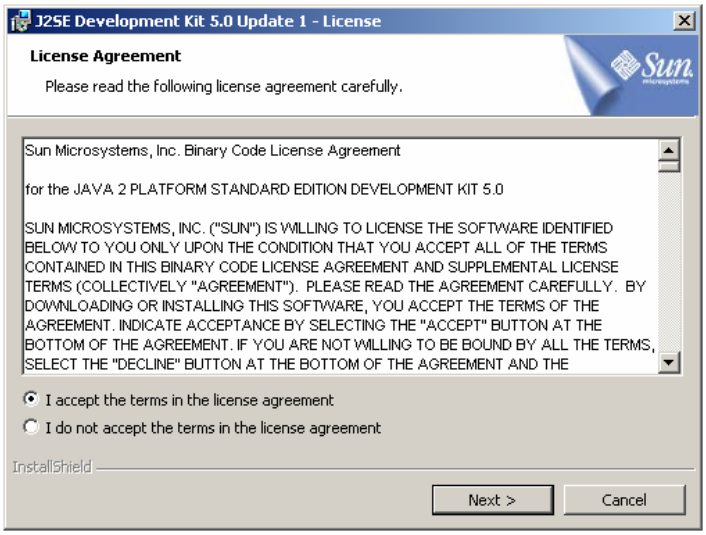

J2SE Development Kit 5.0 Update 1 - License
License Agreement
Please read the following license agreement carefully.
Sun Microsystems, Inc. Binary Code License Agreement
for the JAVA 2 PLATFORM STANDARD EDITION DEVELOPMENT KIT 5.0
SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THE SOFTWARE IDENTIFIED BELOW TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY CODE LICENSE AGREEMENT AND SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT"). PLEASE READ THE AGREEMENT CAREFULLY. BY DOWNLOADING OR INSTALLING THIS SOFTWARE, YOU ACCEPT THE TERMS OF THE AGREEMENT. INDICATE ACCEPTANCE BY SELECTING THE "ACCEPT" BUTTON AT THE BOTTOM OF THE AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY ALL THE TERMS, SELECT THE "DECLINE" BUTTON AT THE BOTTOM OF THE AGREEMENT AND THE
I accept the terms in the license agreement
I do not accept the terms in the license agreement
InstallShield
Next >
Cancel

Figura 76 - Tela de instalação do JAVA

Selecionar a opção “I accept ...” e logo em seguida clicar no botão Next > conforme a Figura 76.

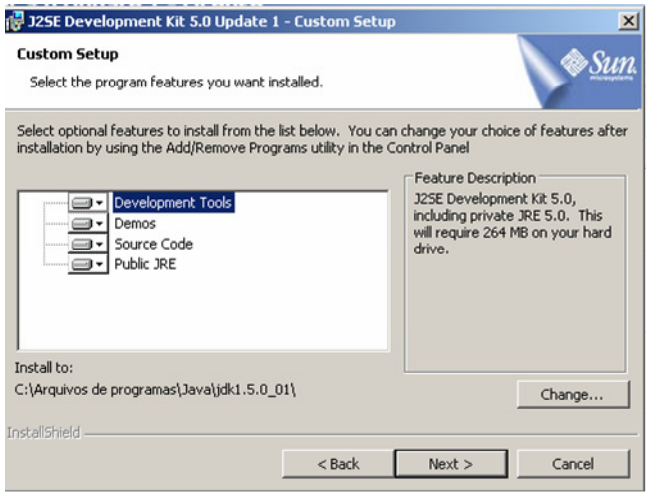

J2SE Development Kit 5.0 Update 1 - Custom Setup
Custom Setup
Select the program features you want installed.
Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel
Development Tools
Demos
Source Code
Public JRE
Feature Description
J2SE Development Kit 5.0, including private JRE 5.0. This will require 264 MB on your hard drive.
Install to:
C:\Arquivos de programas\Java\jdk1.5.0_01\
Change...
InstallShield
< Back
Next >
Cancel

Figura 77 - Tela de personalização da instalação do JAVA

Clicar no botão Next > , conforme a Figura 77.

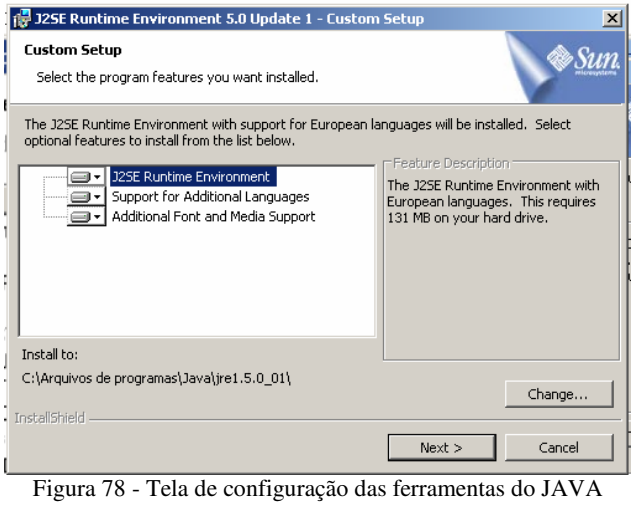

J2SE Runtime Environment 5.0 Update 1 - Custom Setup
Custom Setup
Select the program features you want installed.
The J2SE Runtime Environment with support for European languages will be installed. Select optional features to install from the list below.
J2SE Runtime Environment
Support for Additional Languages
Additional Font and Media Support
Feature Description
The J2SE Runtime Environment with European languages. This requires 131 MB on your hard drive.
Install to:
C:\Arquivos de programas\Java\jre1.5.0_01\
Change...
InstallShield
Next >
Cancel

Figura 78 - Tela de configuração das ferramentas do JAVA

Clicar no botão Next > , conforme a Figura 78.

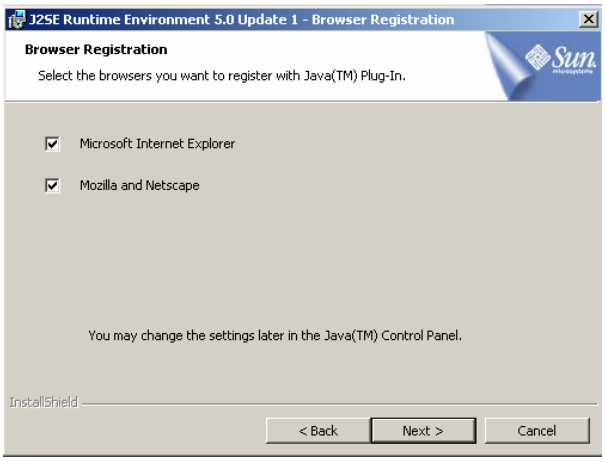

J2SE Runtime Environment 5.0 Update 1 - Browser Registration
Browser Registration
Select the browsers you want to register with Java(TM) Plug-In.
Microsoft Internet Explorer
Mozilla and Netscape
You may change the settings later in the Java(TM) Control Panel.
InstallShield
< Back
Next >
Cancel

Figura 79 - Tela de configuração do browser

Clicar no botão Next > , conforme a Figura 79.

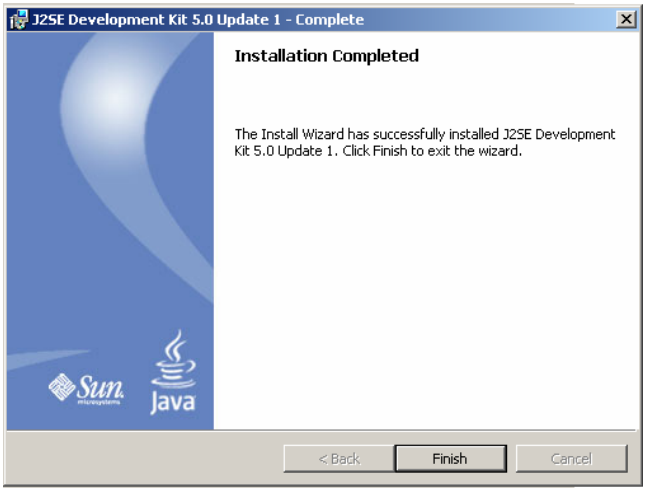

J2SE Development Kit 5.0 Update 1 - Complete
Installation Completed
The Install Wizard has successfully installed J2SE Development Kit 5.0 Update 1. Click Finish to exit the wizard.
Sun
Java
< Back
Finish
Cancel

Figura 80 - Tela de finalização da instalação

Clicar no botão Finish, conforme a Figura 80.

A instalação do Java está concluída. A seguir será necessária a configuração das variáveis de ambiente para seu correto funcionamento.

2.1 Variáveis de Ambiente

Após a instalação do JDK, será necessário criar a variável de ambiente JAVA_HOME, necessária para instalação do eclipse. No Windows XP, para criar e alterar as variáveis de ambiente é necessário um clique com o botão direito do mouse sobre o ícone “Meu Computador”. Aparecerá um menu suspenso, onde deve-se clicar em “propriedades”, conforme a Figura 81.

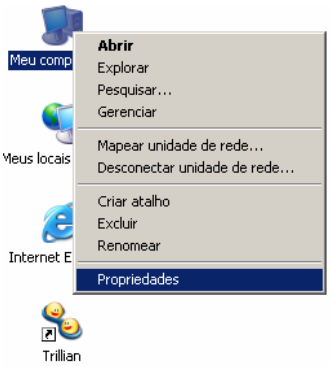


Figura 81 - Configuração das variáveis de Ambiente 1

Aparecerá uma tela, clicar na aba Avançado, conforme a Figura 82.

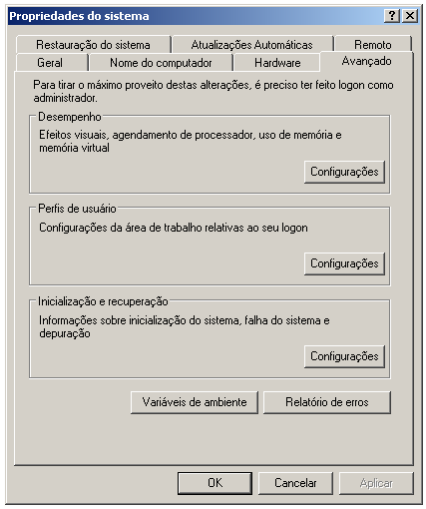


Figura 82 - Configuração das variáveis de Ambiente 2

Clicar no botão “Variáveis de Ambiente”, e aparecerá uma tela conforme Figura 83.

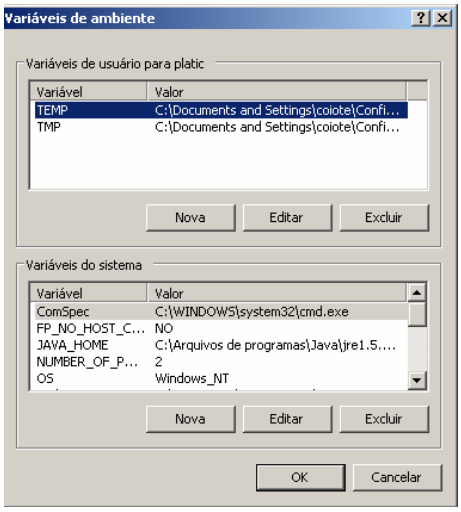


Figura 83 - Configuração das variáveis de Ambiente 3

Abaixo de “Variáveis do sistema” temos os botões “Nova”, “Editar” e “Excluir”, conforme Figura 83. Será usado o botão “Nova” para criar a nova variável JAVA_HOME e “Editar” para alterar as variáveis PATH e PATHEXT.

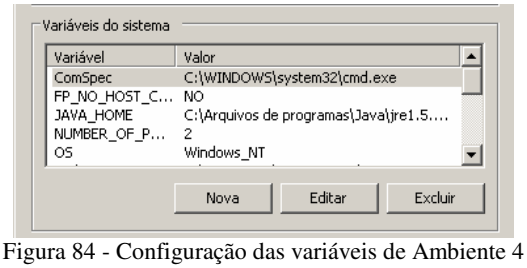


Figura 84 - Configuração das variáveis de Ambiente 4

Será necessário editar a variável PATH, adicionando o caminho do kit de desenvolvimento JAVA, para isso localizar a pasta bin (é a que contém os aplicativos java, javac e etc) aonde foi instalado o JDK. Por exemplo:

"C:\Arquivos de programas\Java\jdk1.5.0_01\bin"

Após seguiu os passos descritos no primeiro parágrafo do item 2.1, selecionar a variável PATH e em seguida clicar no botão Editar, conforme Figura 85.

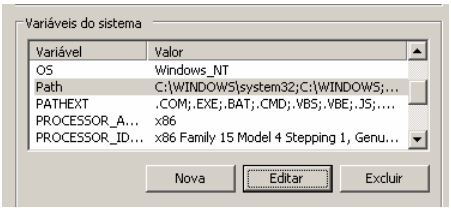


Figura 85 - Configuração da variável PATH 1

No campo valor da variável adicionar, ao final, um ponto e vírgula "**;**" e copiar o caminho da pasta bin onde está instalado o jdk, conforme Figura 86.

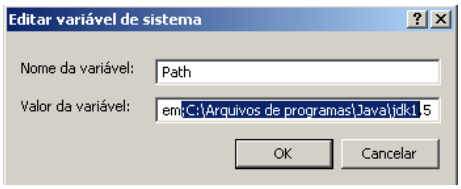


Figura 86 - Configuração da variável PATH 2

Adicionar ao PATHEXT o ".JAR" para que o sistema possa reconhecer os arquivos JAR. Para isso selecionar a variável PATHTEXT, clicar no botão "Editar". No campo valor da variável, adicionar, ao final, um ponto e vírgula "**;**" e digitar "JAR" sem as aspas.

Criar a variável de ambiente "JAVA_HOME" e adicionar o caminho onde está instalado o JDK. Por exemplo:

"C:\Arquivos de programas\Java\jdk1.5.0_01"

Clicar no botão "Nova", conforme Figura 87.

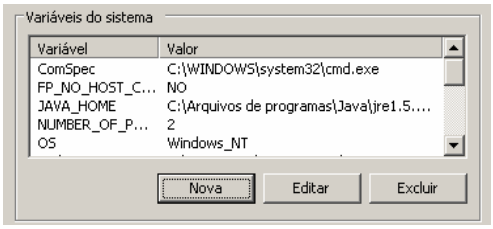


Figura 87 - Criar a variável JAVA_HOME 1

No campo "Nome da Variável" colocar JAVA_HOME, e no campo valor da variável colocar o caminho onde se encontra instalado o jdk, conforme figura 48.

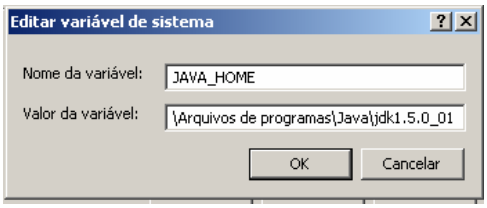


Figura 88 - Criar a variável JAVA_HOME 2

O processo de instalação do JAVA está concluído.

3 Instalação TOMCAT

IMPORTANTE: Para a instalação do TOMCAT 5.5 é necessário que o JDK 1.5.0 ou superior esteja instalado e que a variável JAVA_HOME esteja corretamente configurada.

O primeiro passo é baixar o arquivo de instalação do Tomcat. Disponível em: http://jakarta.apache.org/

Após efetuar o *download* do arquivo, executar o arquivo de instalação do Tomcat. Clicar no botão Next >, conforme a Figura 89.

Apache Tomcat Setup
Welcome to the Apache Tomcat Setup Wizard
This wizard will guide you through the installation of Apache Tomcat.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
The Apache Jakarta Project
http://jakarta.apache.org/
Next >
Cancel

Figura 89 - Tela de inicio da instalação do Tomcat

Clicar no botão "*I Agree*" para continuar a instalação, conforme Figura 90.

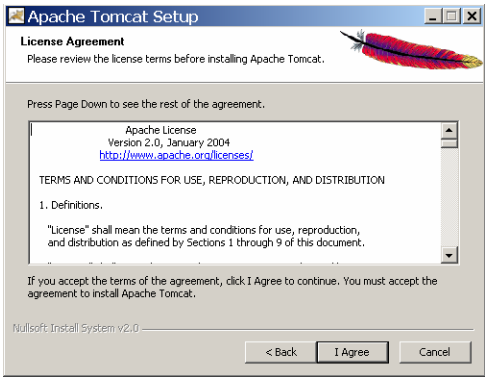

Apache Tomcat Setup
License Agreement
Please review the license terms before installing Apache Tomcat.
Press Page Down to see the rest of the agreement.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.
"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Apache Tomcat.
Nullsoft Install System v2.0
< Back
I Agree
Cancel

Figura 90 - Termo de licença do Tomcat

Escolher a opção “full” e clicar em Next>, conforme Figura 91.

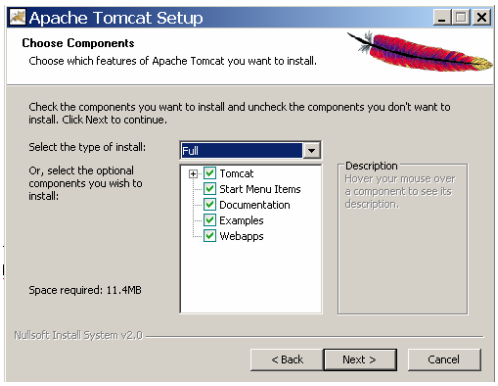

Apache Tomcat Setup
Choose Components
Choose which features of Apache Tomcat you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select the type of install:
Full
Or, select the optional components you wish to install:
Tomcat
Start Menu Items
Documentation
Examples
Webapps
Description
Hover your mouse over a component to see its description.
Space required: 11.4MB
Nullsoft Install System v2.0
< Back
Next >
Cancel

Figura 91 - Configuração do Tomcat

Escolher o diretório a onde será instalado o Tomcat e clicar no botão Next>, conforme Figura 92.

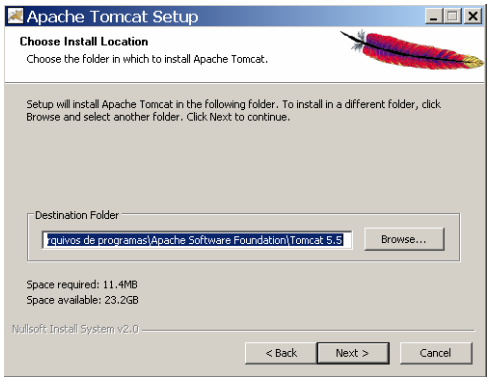

Apache Tomcat Setup
Choose Install Location
Choose the folder in which to install Apache Tomcat.
Setup will install Apache Tomcat in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
rquivos de programas\Apache Software Foundation\Tomcat 5.5
Browse...
Space required: 11.4MB
Space available: 23.2GB
Nullsoft Install System v2.0
< Back
Next >
Cancel

Figura 92 - Configuração do local de instalação do Tomcat 1

Configurar a porta utilizada pelo Tomcat, geralmente na porta 8080, definir o login e senha, conforme a figura 53.

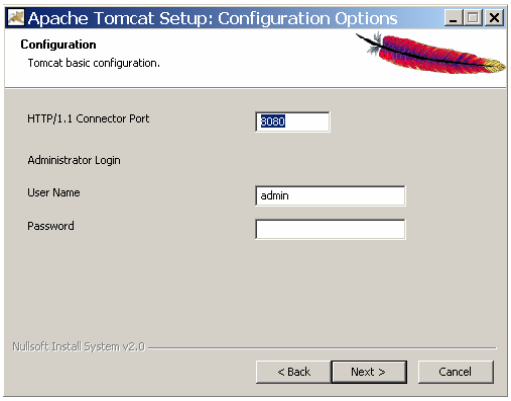


Figura 93 - Tela de configuração do Tomcat 2

Verificar o diretório é o mesmo onde está instalado o JAVA e clicar no botão “Install”, conforme figura 54.

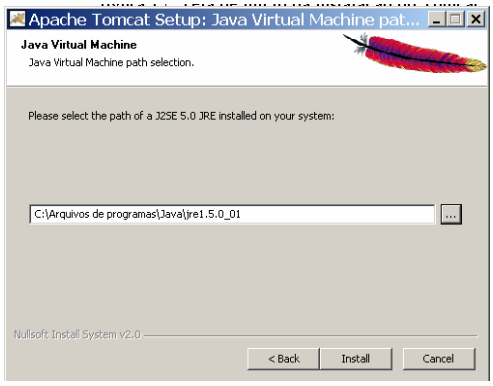

Apache Tomcat Setup: Java Virtual Machine pat...
Java Virtual Machine
Java Virtual Machine path selection.
Please select the path of a J2SE 5.0 JRE installed on your system:
C:\Arquivos de programas\Java\jre1.5.0_01
...
Nullsoft Install System v2.0
< Back
Install
Cancel

Figura 94 - Tela de configuração do Java no Tomcat

Clicar no botão “Finish” para terminar a instalação, conforme figura 55.

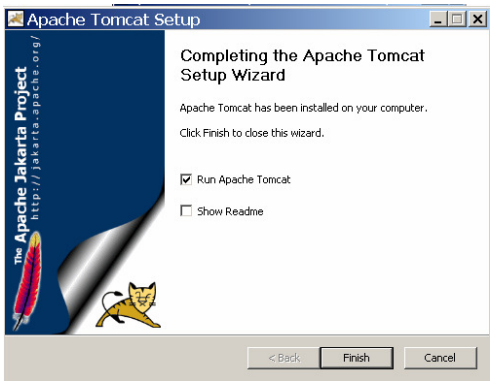


Figura 95 - Tela de termino da instalação do Tomcat

Verificar se o Tomcat sendo executado. Abrir o *browser* e digitar http://127.0.0.1:8080/ na barra de endereços; Irá aparecer a página principal do Tomcat, conforme a figura 56.

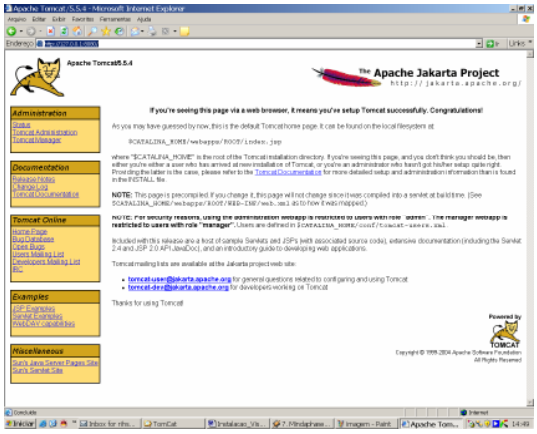


Figura 96 - Pagina principal do Tomcat

Após este processso, o Tomcat já está instalado e operacional.

4 Instalação do MySQL

O primeiro passo é baixar o arquivo de instalação do MySQL. Disponível em: http://dev.mysql.com/downloads/

Após o *download*, executar o arquivo de instalação do MySql. Na primeira tela, clicar no botão Next, conforme Figura 97.

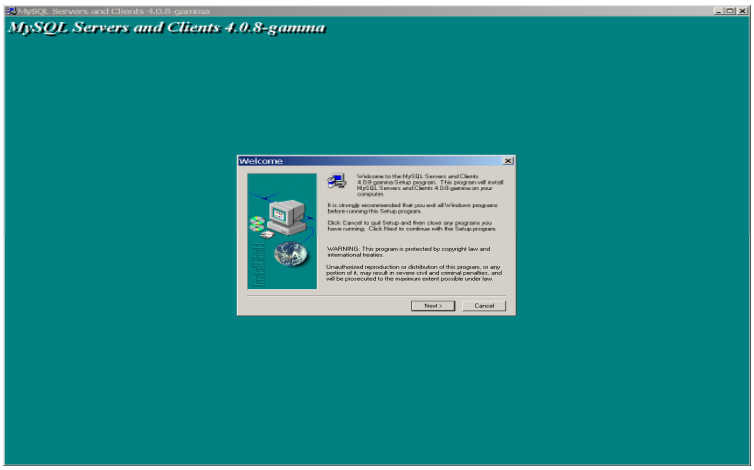


Figura 97 - Inicio da instalação do MySQL

Clicar em Next, conforme a Figura 98.

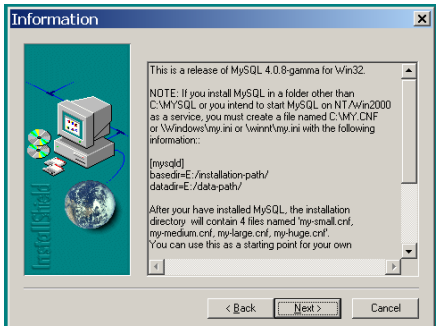

Information
This is a release of MySQL 4.0.8-gamma for Win32.
NOTE: If you install MySQL in a folder other than C:\MYSQL or you intend to start MySQL on NT/Win2000 as a service, you must create a file named C:\MY.CNF or \Windows\my.ini or \winnt\my.ini with the following information::
[mysqld]
basedir=E:/installation-path/
datadir=E:/data-path/
After your have installed MySQL, the installation directory will contain 4 files named 'my-small.cnf, my-medium.cnf, my-large.cnf, my-huge.cnf'.
You can use this as a starting point for your own
< Back
Next >
Cancel

Figura 98 - Tela de informação do MySQL

Para escolher o diretório de instalação clicar no botão Browser, e depois clicar em Next>, conforme Figura 99.

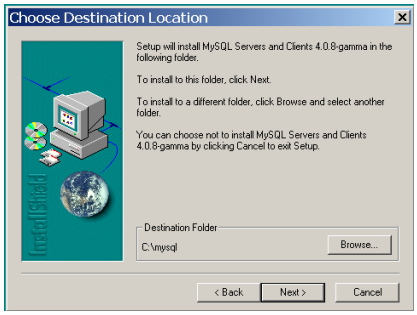

Choose Destination Location
Setup will install MySQL Servers and Clients 4.0.8-gamma in the following folder.
To install to this folder, click Next.
To install to a different folder, click Browse and select another folder.
You can choose not to install MySQL Servers and Clients 4.0.8-gamma by clicking Cancel to exit Setup.
Destination Folder
C:\mysql
Browse...
< Back
Next >
Cancel

Figura 99 - Escolha do diretório do MySQL

Escolher a opção Typical e clicar no botão Nest>, conforme Figura 100.

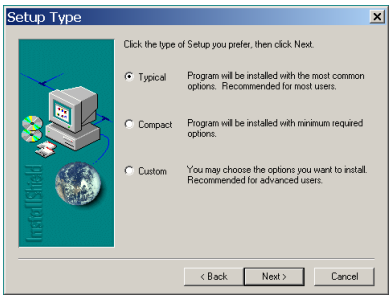

Setup Type
Click the type of Setup you prefer, then click Next.
Typical
Program will be installed with the most common options. Recommended for most users.
Compact
Program will be installed with minimum required options.
Custom
You may choose the options you want to install. Recommended for advanced users.
< Back
Next >
Cancel

Figura 100 - Configuração do MySQL

Clicar no botão Finish, conforme Figura 101.

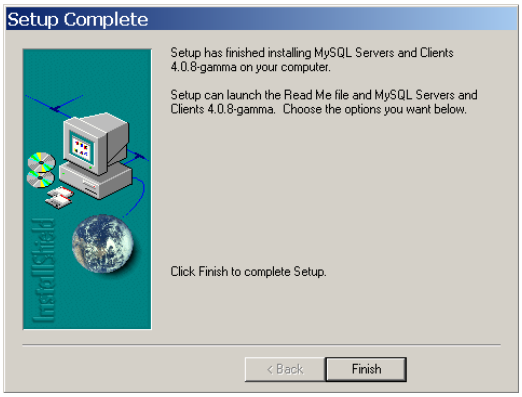

Setup Complete
Setup has finished installing MySQL Servers and Clients 4.0.8-gamma on your computer.
Setup can launch the Read Me file and MySQL Servers and Clients 4.0.8-gamma. Choose the options you want below.
Click Finish to complete Setup.
< Back
Finish

Figura 101 - Termino da instalação do MySQL

Para iniciar o MySQL, deve-se ir na pasta BIN que se encontra no diretório onde foi instalado o MySQL (C:\mysql\bin), e executar o arquivo “winmysqladmin.exe” para inicializar o MySQL, conforme Figura 102.

Figura 102 - Executar o MySQL

Após a execução deste processo, o MySQL deverá estar executando e operacional.

5 Instalação do driver ODBC do MySql.

O primeiro passo é baixar o arquivo de instalação do driver ODBC do MySQL. Disponível em:

http://dev.mysql.com/downloads/

Após o *download*, executar o arquivo de instalação do MySql (MyODBC-3.51.11-2-win). Clique no botão Next, conforme figura 63.

MyODBC Setup
Welcome to the MyODBC Installation Wizard
It is strongly recommended that you exit all Windows programs before running this setup program.
Click Cancel to quit the setup program, then close any programs you have running. Click Next to continue the installation.
WARNING: This program is protected by copyright law and international treaties.
Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.
MySQL
< Back
Next >
Cancel

Figura 103 - Executar o MySQL

Selecionar a opção I Agree e clicar no botão Next, conforme Figura 104.

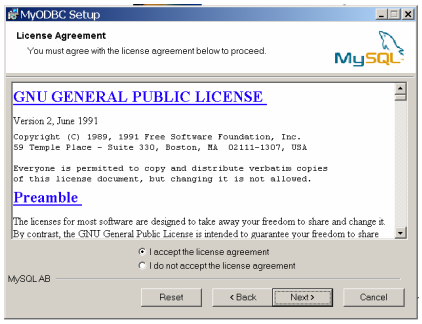

MyODBC Setup
License Agreement
You must agree with the license agreement below to proceed.
MySQL
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
Preamble
The licenses for most software are designed to take away your freedom to share and change it.
By contrast, the GNU General Public License is intended to guarantee your freedom to share
I accept the license agreement
I do not accept the license agreement
MySQL AB
Reset
< Back
Next >
Cancel

Figura 104 - Licença do *driver* ODBC do MySQL

Clicar no botão Next>, conforme Figura 105.

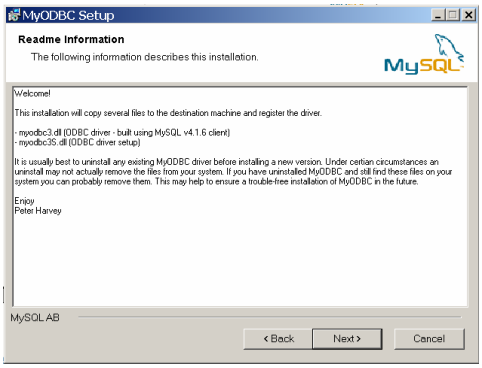

MyODBC Setup
Readme Information
The following information describes this installation.
MySQL
Welcome!
This installation will copy several files to the destination machine and register the driver.
- myodbc3.dll (ODBC driver - built using MySQL v4.1.6 client)
- myodbc3S.dll (ODBC driver setup)
It is usually best to uninstall any existing MyODBC driver before installing a new version. Under certain circumstances an uninstall may not actually remove the files from your system. If you have unnstalled MyODBC and still find these files on your system you can probably remove them. This may help to ensure a trouble-free installation of MyODBC in the future.
Enjoy
Peter Harvey
MySQL AB
< Back
Next >
Cancel

Figura 105 - Tela de boas vindas do ODBC MySQL

Clicar no botão Next>, conforme Figura 106.

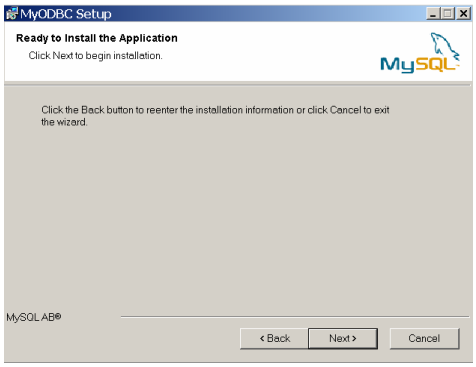

MyODBC Setup
Ready to Install the Application
Click Next to begin installation.
MySQL
Click the Back button to reenter the installation information or click Cancel to exit the wizard.
MySQL AB®
< Back
Next >
Cancel

Figura 106 - Tela de confirmação

Clicar no botão Finish, conforme a figura 67.

Figura 107 - Tela de finalização da instalação do ODBC MySQL

A próxima etapa é configurar o ODBC. Para isso, ir no painel de controle, em ferramentas administrativas, e abrir i item Fonte de Dados (ODBC), conforme Figura 108.

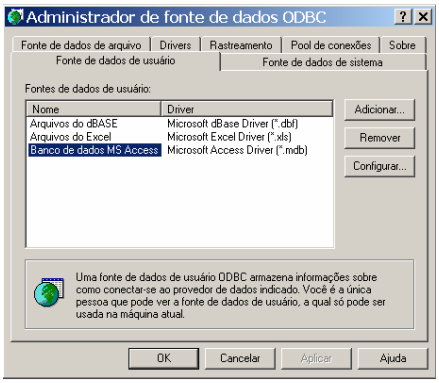

Administrador de fonte de dados ODBC
Fonte de dados de arquivo
Drivers
Rastreamento
Pool de conexões
Sobre
Fonte de dados de usuário
Fonte de dados de sistema
Fontes de dados de usuário:
Nome
Driver
Arquivos do dBASE
Microsoft dBase Driver (*.dbf)
Arquivos do Excel
Microsoft Excel Driver (*.xls)
Banco de dados MS Access
Microsoft Access Driver (*.mdb)
Adicionar...
Remover
Configurar...
Uma fonte de dados de usuário ODBC armazena informações sobre como conectar-se ao provedor de dados indicado. Você é a única pessoa que pode ver a fonte de dados de usuário, a qual só pode ser usada na máquina atual.
OK
Cancelar
Aplicar
Ajuda

Figura 108 - Adicionando a fonte de dados MySQL 1

Na aba Fonte de Dados clicar no botão Adicionar. Selecionar a Fonte de dado MySQL e clicar no botão concluir, conforme figura 69.

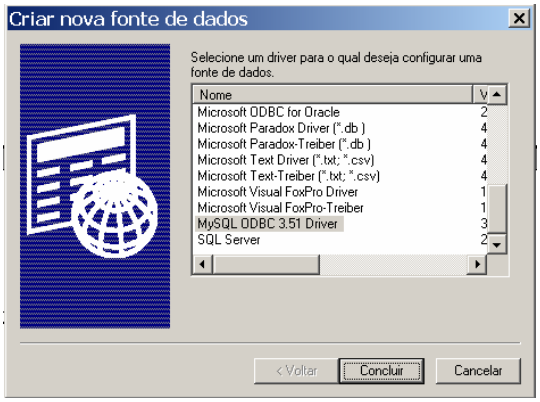


Figura 109 - Adicionando a fonte de dados MySQL 2

Preencher os dados conforme a figura 70 e clicar em “Ok”. É possível testar a configuração clicando no botão Test.

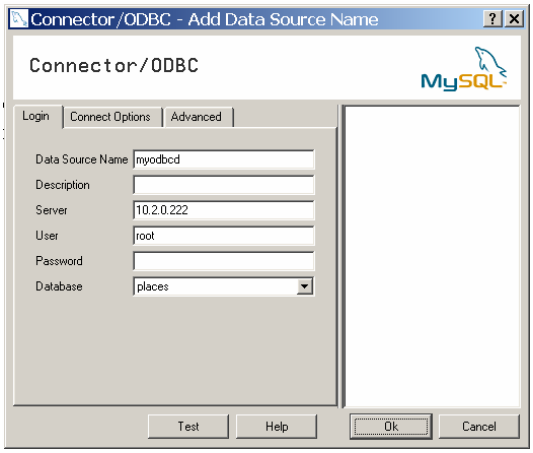


Figura 110 - Adicionando a fonte de dados MySQL 3

Na aba Fonte de Dados do sistema clicar no botão Adicionar e siguir os passos anteriores descritos a partir da aba Fonte de Dados. Com a execução deste processo, o acesso ao MySQL pode ser feito via ODBC.

6 Instalação do Eclipse

IMPORTANTE: Para a instalação do Eclipse é necessário que o JDK esteja instalado e que a variável JAVA_HOME, PATH e PATHTEXT estejam corretamente configurados conforme item 2 deste anexo.

O primeiro passo é baixar o arquivo de instalação do Eclipse. Disponível em: http://www.eclipse.org/downloads/. O Eclipse não possui um processo de instalação, basta descompactar o arquivo baixado em uma pasta qualquer. Na pasta Eclipse terá um arquivo eclipse. Ao executar este arquivo, o eclipse será iniciado. Criar um atalho deste arquivo para executar o eclipse a partir da área de trabalho do usuário.

ANEXO 2 – REQUISITOS DO MÓDULO DE GERÊNCIA DE REQUISITOS

Requisitos Funcionais (REF)

REF.01 O sistema deve permitir que um operador possa efetuar o login no sistema.

REF.02 O sistema deve permitir o cadastro dos requisitos.

REF.03 O sistema deve permitir a criação de associações entre os requisitos para prover rastreabilidade.

REF.04 O sistema deve permitir a atribuição de responsabilidade dos requisitos a algum participante do projeto.

REF.05 O sistema deve permitir o cadastro de *baselines*.

REF.06 O sistema deve permitir a inclusão de requisitos em uma *baseline*.

REF.07 O sistema deve mostrar a rastreabilidade entre os requisitos através de uma matriz de requisitos.

REF.08 O sistema deve permitir anexar documentos externos (arquivos) ao projeto.

Requisitos Não Funcionais (RNF)

Segurança

RNF.01.01 O sistema deve possuir um mecanismo de controle de segurança para evitar que pessoas não autorizadas tenham acesso ao sistema

RNF.01.02 As senhas cadastradas que estiverem no banco de dados não deverão ser visíveis diretamente, devendo estar em um modo criptografado.

Usabilidade

RNF.02.01 Os navegadores deverão ser compatíveis com o Internet Explorer 6.0 ou versões superiores.

Confiabilidade

RNF.03.01 O sistema deverá ser suficientemente robusto para permitir acessos 24h por dia, todos os dias da semana.

Desempenho

RNF.04.01 O sistema deve estar preparado para atender até 5 usuários simultâneos.

Software e Hardware

RNF.05.01 O sistema deve estar preparado para trabalhar com o banco de dados MYSQL 4.1 ou superior.

RNF.05.02 O sistema será executado em rede de computadores com protocolo TCP/IP.

RNF.05.03 O sistema será executado em um computador que tenha acesso a Internet.

RNF.05.04 O sistema será executado em um computador que tenha no mínimo 512 MB de memória RAM e 1GB de espaço em disco

8 BIBLIOGRAFIA

ALLUR, Deepak. **Core J2EE Patterns Best Practices na Disign Strategies**. São Paulo: SUN Press,2002 – 2 ed. ISBN: 0131422454

ANDRADE, Maria Margarida de. **Introdução à metodologia do trabalho científico.** São Paulo: Atlas, 1999 - 4 ed. 153 p. ISBN: 85-224-2300-8

AURÉLIO (Aurélio Buarque de Holanda Ferreira). **Dicionário Aurélio da Língua Portuguesa**. 2. ed., Nova fronteira.,1999. 2128 p. ISBN 85-209-0411-4.

ANACLETO, Alessandra. **Processo e Modelo de Avaliação para Melhoria de Processos de Software em Micro e Pequenas Empresas**. Trabalho TI da Pós-Graduação da Universidade Federal de Santa Catarina, 2004.

BACCA, Claudia. **Project's Manager Spotlight on Change Management**. San Francisco: Harbor Light Press, 2005

BAUER, Christian; KING, Gavin. ***Hibernate in action.*** Greenwich: MANNING. 2005. 432 p. ISBN 1932394-15-X.

CERVO, A.L.; BERVIZ, P. A. **Metodologia científica.** 4. ed. São Paulo: Makron Books, 1996. 209 p. ISBN 85-346-0521-1.

CORDEIRO, Marco Aurélio. **Uma ferramenta automatizada de suporte ao processo de gerenciamento de requisitos**. Dissertação de Mestrado, Pontifica Universidade Católica do Paraná. Curitiba, 2002.

FIORINI, Soeli; STAA, Arndt; BAPTISTA, Renan. **Engenharia de Software com CMM**. Rio de Janeiro: Brasport, 1999. 341 p. ISBN 85-85840-84-6.

GIL, Antônio Carlos. **Métodos e técnicas de pesquisa social**. 5.ed. São Paulo: Atlas, 1999. 206 p. ISBN 85-224-2270-2.

HAZAN, Claudia. Tutorial sobre **Implantação de um Processo de Gestão de Requisitos seguindo o CMMI**. São Paulo. SIMPROS (Simpósio Internacional de Melhoria de Processo de Software), 2004.

HUSTED, Ted. **Struts em Ação**. São Paulo: Ciência Moderna,2004. 632p. ISBN: 8573932996

Institute of Electrical and Electronic Engineers (IEEE). **IEEE Standard Glossary of Software Engineering Terminology**. New York, NY: Institute of Electrical and Electronic Engineers, 1983.

INTERNATIONAL ORGANIZATION OF STANDARIZATION (ISO). **ISO/IEC TR 9000 2000 Standards**. Genebra, 2000. Disponível em <http://www.iso.org>. Acesso em 7 de agosto de 2005.

INTERNATIONAL ORGANIZATION OF STANDARIZATION (ISO). **ISO/IEC TR 15504** : *Information Technology - Software Process Assessment*. Genebra, 2003. Disponível em <http://www.iso.org>. Acesso em 7 de agosto de 2005.

INTERNATIONAL ORGANIZATION OF STANDARIZATION (ISO). **ISO/IEC TR 12207:** *Information Technology - Software Life Cycle Processes*. Genebra, 1995. Disponível em <http://www.iso.org>. Acesso em 7 de agosto de 2005.

JACOBSON, Ivar *et al*. **Object-Oriented Software Engineering**. Harlow: Addison-Wesley, 1998. 528 p. ISBN 0-201-54435-0.

KOTONYA, G; SOMMERVILLE, I. **Requirements Engineering: Process and Tecniques**. John Wiley & Sons, 1998.

LEITE, Julio Cesar S. P.; SAYAO, Mirian. **Rastreabilidade de Requisitos**. Relatório Técnico 20/05, série Monografias em Ciência da Computação. Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). 2005.

MCT. **Ministério de Ciência e Tecnologia**. Disponível em: <http://www.mct.gov.br >. Acesso em 15 de março de 2005.

MORETTO, Luís Augusto Machado. **Places: uma plataforma colaborativa para engenharia de software via web**. 2005. Paginação irregular. Trabalho de conclusão de curso (Bacharelado em Ciência da Computação) – Curso de Ciência da Computação, Universidade do Vale do Itajaí- Univali, São José, São José, 2005.

SOUZA, Richard Henrique de. **PLACES: Um módulo de estimativa de tamanho baseado em pontos de caso de uso**. 2005. Paginação irregular. Trabalho de conclusão de curso

(Bacharelado em Ciência da Computação) – Curso de Ciência da Computação, Universidade do Vale do Itajaí- Univali, Centro de Educação São José, São José, 2005.

PADUAN, Roberta. **A volta da qualidade**. Revista EXAME, 15 de janeiro de 2003. Pág. 64 a 67.

PAETSCH, Flauke - **Requirements Engineering In Agile Software Development.** Dissertação de Mestrado – Universidade de Calgary, Canadá, 2003.

PFLEEGER, Shari Lawrence - **Engenharia de Software.** 3 ed. Prentice-Hall, 1998. ISBN 85-879-1831-1.

PRESSMAN, Roger S. **Engenharia de Software**. trad. José Carlos Barbosa dos Santos. São Paulo: Makron Books, 1995. 1056 p. ISBN: 85-346-0237-9

RAIBLE,Matt. **Spring Liveby**. LLC.Cover: SourceBeat. 2004. 160 p. ISBN: 0974884340

REINEHR, Sheila. **Curso de Gerenciamento de Requisitos CMM/CMMI**. Florianópolis: CTAI/SENAI, 2006.

RICHARDSON, Roberto Jarry et al**. Pesquisa Social: Métodos e Técnicas**. 3 ed. São Paulo: Atlas S.A., 1999. 334 p. ISBN 85-224-2111-0.

ROCHA, Ana Regina; MALDONADO, José Carlos; WEBER, Kival. **Qualidade de Software – Teoria e Prática**. São Paulo: Prentice Hall, 2001. 277 p. ISBN 85-87918-54-0.

SAYÃO, Mirian; BREITMAN, Karin, LEITE, Julio Cesar S. P. **Tutorial sobre Gerencia de Requisitos**. Brasília, Simpósio Brasileiro de Engenharia de Software (SBES). 2004.

SILVA, Edna Lúcia; MENEZES, Estera Muszkat. **Metodologia da pesquisa e elaboração de dissertação.** 3. ed. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001.

SOCIEDADE PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO (SOFTEX). **mps.BR**. Campinas, 2005. Disponível em <http://www.softex.br/mpsbr>. Acesso em 7 de agosto de 2005.

SOFTWARE ENGINEERING INSTITUTE (SEI). **Capability Maturity Model (CMM)**. Pittsburgh, 1998. Disponível em: <http://www.sei.cmu.edu/cmm/>. Acesso em: 7 de agosto de 2005.

SOFTWARE ENGINEERING INSTITUTE (SEI). **Capability Maturity Model Integration**, **Version 1.1**. Pittsburgh, 2002. Disponível em: <http://www.sei.cmu.edu/cmmi/>. Acesso em: 7 de agosto de 2005.

SOMMERVILLE, Ian. **Software endineering.** 6 ed. Harlow. Addison-Wesley, 2001.

SUN. **Java**. Agosto, 2006. http://java.sun.com

WALLS, Craig; BREIDENBACH, Ryan. ***Spring in action.*** Greenwich: MANNING. 2005. 472 p. ISBN 1-932394-35-4.

WEBER, Kival Chaves. **Qualidade e produtividade em software**. 4. ed., renovada. São Paulo: Makron, 2001.

WIEGERS, K. E. **Software Requirements**. Microsoft Press, 1999.

9 ASSINATURAS

_____

Eduardo Miller
Acadêmico

_____

Marcello Thiry
Professor Orientador